НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.049

«До захисту допущено»

Завідувач кафедри
_____ І.Р. Пархомей
(підпис)
“___”_____2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 126 «Інформаційні системи та технології»

на тему: Data Mining та машинні техніки навчання для виявлення вторгнення в кібербезпеку робототехнічних та автономних систем

Виконав: студент другого курсу, групи ІК-72мп
(шифр групи)

_____Петрухно І.Р._____ _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник доцент, к.т.н., доцент Бурлаков В.М_____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент_____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ І.Р. Пархомей
(підпис)
«___»_____2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Студенту Петрухну Ігорю Руслановичу
(прізвище, ім'я, по батькові)

1. Тема дисертації Data Mining та машинні техніки навчання для виявлення вторгнення в кібербезпеку робототехнічних та автономних систем
науковий керівник дисертації__доцент, к.т.н., доцент Бурлаков В.М._____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «___»_____ 2018 р. №_____

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження – система IDS на базі методології паралельних обчислень використовуючи інструменти Hadoop.

4. Предмет дослідження – методи та процеси Data Mining і машинних технік навчання для виявлення вторгнення в кібербезпеку робототехнічних та автономних систем.

5. Перелік завдань, які потрібно розробити – аналіз існуючих методів та технік Data Mining; аналіз існуючих систем Data Mining в Кібербезпеці; розробка програмного забезпечення на базі Hadoop; дослідження ефективності алгоритму MapReduce на базі Hadoop.

6. Орієнтовний перелік ілюстративного матеріалу – п'ять плакатів

7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	13.09.2018 р.	
2	Постановка задачі	15.09.2018 р.	
3	Аналіз інформаційного забезпечення	20.09.2018 р.	
5	Аналіз алгоритмічного забезпечення	25.09.2018 р.	
6	Розробка алгоритмічного забезпечення	15.10.2018 р.	
7	Розробка програмного забезпечення	01.11.2018 р.	
8	Маркетинговий аналіз стартап-проекту	10.11.2018 р.	
9	Висновки	15.11.2018 р.	

Студент _____ Петрухно І.Р.
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Бурлаков В.М.
(підпис) (ініціали, прізвище)

АНОТАЦІЯ

У роботі розглянуто проблему в області кібербезпеки повязану з методами аналізу великих масивів даних для робототехнічних систем. Об’єктом даної роботи є дослідницька система на базі методології паралельних обчислень використовуючи інструменти Hadoop.

Предметом виступають методи та процеси Data Mining і машинних технік навчання для виявлення вторгнення в кібербезпеку робототехнічних і автономних систем.

В поданої роботі, розглянуто основні особливості існуючої системи (SIEM). які дозволяє оброблювати великі масиви даних, її переваги та недоліки, Здійснений аналіз тактик по побудові Security Analitics System, які впливають на точність, надійність, продуктивність, масштабованість проектуємих IDS систем.

Реалізована дослідницька система на базі методології паралельних обчислень використовуючи інструменти Hadoop, що забезпечує ефективне функціонування в умовах атак.

Дана система може бути використана в діяльності конкретної установи, а також може бути використаний і іншими установами для вдосконалення паралельних обчислень використовуючи інструменти Hadoop, також дана концепція викладу даного дослідження може бути використана в якості методичного посібника при розробці системи виявлення вторгнення в кібербезпеку робототехнічних і автономних систем.

Дозволяє збільшити швидкість обробки даних та зменшити час аналізу данних використовуючи парадигму MapReduce.

Ключові слова: IDS, Data Mining, Hadoop, SIEM, MapReduce, CyberSecurty, Robot defence.

Розмір пояснювальної записки – 111 аркушів, містить 31 ілюстрацій, 26 таблиць, 5 додатків.

ABSTRACT

The paper deals with the problem of cybersecurity associated with methods of analysis of large data sets for robotic systems. The object of this work is a research system based on the methodology of parallel computing using Hadoop tools.

The subject is the methods and processes of Data Mining and machine learning techniques to detect the invasion of the cybersecurity of robotic and autonomous systems.

In the given work, the main features of the existing system (SIEM) are considered. which allows processing large volumes of data, its advantages and disadvantages, Analysis of the tactics for constructing the Security Analitics System, which affect the accuracy, reliability, performance, scalability of project IDS systems.

A research system implemented on the basis of parallel computing methodology using the Hadoop tools, which provides effective operation under attack conditions.

This system can be used in the activities of a particular institution, and can also be used by other institutions to improve parallel computing using Hadoop tools, this concept can also be used as a methodological guide for the development of a system for detecting cybersecurity robotic and autonomous systems .

Allows you to increase the speed of data processing and reduce the time of data analysis using the MapReduce paradigm.

Keywords: IDS, Data Mining, Hadoop, SIEM, MapReduce, CyberSecurty, Robot Defense.

The size of the explanatory note is 111 sheets, contains 31 illustrations, 26 tables, 5 appendices.

Пояснювальна записка

до магістерської дисертації

на тему: Data Mining та машинні техніки навчання для виявлення вторгнення в кібербезпеку робототехнічних та автономних систем

Київ – 2018 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АС	—	автоматизована система
ЕОМ	—	електронно-обчислювальна машина
ЗОТ	—	засоби обчислювальної техніки
ІБ	—	інформаційна безпека
ІПКС	—	інцидент/потенційна кризова ситуація
ІР	—	інформаційні ресурси
ІС	—	інформаційна система
ІСАД	—	інтелектуальні системи аналізу даних
КС	—	кризова ситуація
ЛЗ	—	лінгвістична змінна
МНІ	—	машинні носії інформації
МРК	—	мобільних робототехнічних комплексів
ОПР	—	особа, яка приймає рішення
ОС	—	операційна система
ПЗ	—	програмне забезпечення
ПК	—	персональний комп’ютер
ППП	—	пакет прикладних програм
HTT	—	Hyper-Threading Technology
TLP	—	thread-level parallelism

ВСТУП

В цей цифровий вік ми не можемо уявити світ без спілкування. Люди повинні обмінюватися інформацією для різних цілей. Забезпечення зв'язку - величезна проблема через підвищення загроз і атак на мережеву безпеку.

Забезпечення безпеки мережі - головна проблема в епоху інформації з різних типів мережевих загроз і атак. Процес виявлення вторгнень є процесом оцінки підозрілих дій, які відбуваються в мережі. Іншими словами, виявлення вторгнень – це процес ідентифікації і реагування на підозрілу діяльність, спрямовану на обчислювальні або мережеві ресурси. Головне завдання систем виявлення вторгнень (СВВ) полягає в автоматизації функцій по забезпеченню ІБ мережі і забезпечення «прозорості» функцій ІБ для неспеціалістів в області захисту інформації. Тому СВВ – це системи, які збирають інформацію з різних точок мережі (об’єкт, що захищається) і аналізують цю інформацію для виявлення не тільки спроб, але і реальних порушень захисту (вторгнень).

В даний час, на Україні, так і за кордоном ведуться активні роботи зі створення мобільних робототехнічних комплексів (МРК). Сфера застосування комплексів обширна, першочерговими є завдання, в ході яких мобільний робот діє в умовах; небезпечних для знаходження людини. Перспективним є використання автоматизованих робототехнічних комплексів в бойових умовах, коли є пряма загроза життю оператора. У складі пошукової групи МРК можуть здійснювати функції дистанційної розвідки, діючи автономно і передаючи дані по бездротовому каналу.

При розробці МРК одним з найважливіших завдань є забезпечення необхідної умови захищеності інформації. Найбільш уразливими є дані, що передаються через радіоканал. Від поста управління на бортову ЕОМ передаються команди управління, від бортової ЕОМ на пультовій ЕОМ повертаються дані по статусу систем мобільного комплексу та інформація від датчиків (відео камери, радар, приповерхневих сканер і т.д.). Команди, що передаються роботу по бездротовому каналу, можуть бути перехоплені і модифіковані.

Дані, що передаються від мобільного робота на пункт управління, так само можуть бути перехоплені і модифіковані.

До системи аутентифікації в бездротовій мережі пред'являються підвищені вимоги з безпеки. Необхідно використовувати криптографічно стійкі алгоритми, що дозволяють здійснити взаємну аутентифікацію сторін.

РОЗДІЛ 1
АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ ДОСЛІДЖЕННЯ

1.1 Аналіз методів обробки даних застосування Data mining

1.1.1. Апостріорний аналіз даних і журналів мережі

Дані історії мережі та журналу - це дані мережевої активності. Ці дані пасивно контролюються, перевіряються і збираються за допомогою різних механізмів моніторингу, розглядаються в такий спосіб:

Kismet: Kismet - це інструмент сканування, який використовує бездротові детектори 802.11 і дозволяє пасивний моніторинг на основі карт (RF-mon) обнюхувати будь-який стандарт 802.11x мереж. Він відображає протокол ARP (протокол дозволу адрес) і протокол DHCP (протокол динамічної конфігурації хоста) для збереження файлів у форматі файлів Wireshark і TCPDump і відображення рівня активності на різних каналах. Він декодує і вимірює сигнали трафіку в реальному часі. Хакери в основному використовують Kismet, так як він може використовуватися в будь-якій мережі зв'язку. Це допомагає виявити вторгнення. Він працює на платформах Mac і Linux.

Snoop: Sun Microsystems розробили загальний інструмент для виявлення вторгнень Snoop для роботи з платформою Solaris. Він відображає результати в одиночному і багаторядковому форматі. Він прослуховує мережеві пакети IPv4 і IPv6. Цей інструмент схожий на TCPDump при відображенні і форматуванні файлів. Snoop значно хороший, ніж TCPdump завдяки зручному інтерфейсу.

Wireshark: Gerald Combs розробив перший публічний інструмент для обнулення пакетів «Wireshark», раніше відомий як Ethereal. Це аналізатор і аналізатор пакетів з відкритим вихідним кодом і ліцензується GNU GPI (General Public License). Він працює з платформами FreeBSD, UNIX, Linux, Solaris, OpenBSD і Windows. Він зручний для збору, фільтрації та аналізу пакетів. Цей інструмент дуже гнучкий, оскільки його файли журналів знаходяться в іншому форматі.

TCPdump: Національна лабораторія Лоуренса Берклі розробила в 1990 році мережеві засоби сканування і відновлення TCPdump з відкритим вихідним кодом для пакетних мереж TCP / IP (Transmission Control Protocol / Internet Protocol). Користувач перехоплює захоплення і контролює пакети TCP-IP під час передачі в мережі. Він працює з платформами Unix, Linux, Solaris, BSD (Berkeley Software Distribution), Mac і Windows. Він використовує командний рядок для запису і відфільтрування запису в журналі на основі певних правил.

1.1.2. Алгоритми навчання та критеріальні моделі для вивчення апостріорних або журнальних даних

Зібрані дані історії / журналу з мережі вивчаються за допомогою алгоритмів класифікації для побудови моделі прогнозування для ідентифікації хакерів і зловмисників.

Одне з найважливіших призначень методів Data Mining полягає в наочному поданні результатів обчислень, що дозволяє використовувати інструментарій Data Mining людьми, які не мають спеціальної математичної підготовки. У той же час, застосування статистичних методів аналізу даних вимагає доброго володіння теорією ймовірностей і математичної статистики.

Знання, що видобуваються методами Data mining, прийнято представляти у вигляді моделей (рис 1.1).

Рисунок 1.1 – Моделі подання знань Data Mining

Методи побудови таких моделей прийнято відносити до області штучного інтелекту.

Data Mining знайшов широке застосування в науці, дослідженнях, веб-аналітиці, але основне значення і вирішальну роль інтелектуальний аналіз даних має у кібернетиці для виявлення вторгнення в кібербезпеку робототехнічних і автономних систем.

Сьогодні відомі статистичні методи і кібернетичні методи. Перші базуються на вже накопичених знаннях і даних, другі - в основному, на різних математичних підходах.

Статистичні методи Data Mining включають в себе: аналіз вихідних даних, багатомірний статистичний аналіз, аналіз зв'язків і аналіз часових рядів. Кібернетичні методи Data Mining об'єднують методи, засновані на математиці і застосуванні штучного інтелекту.

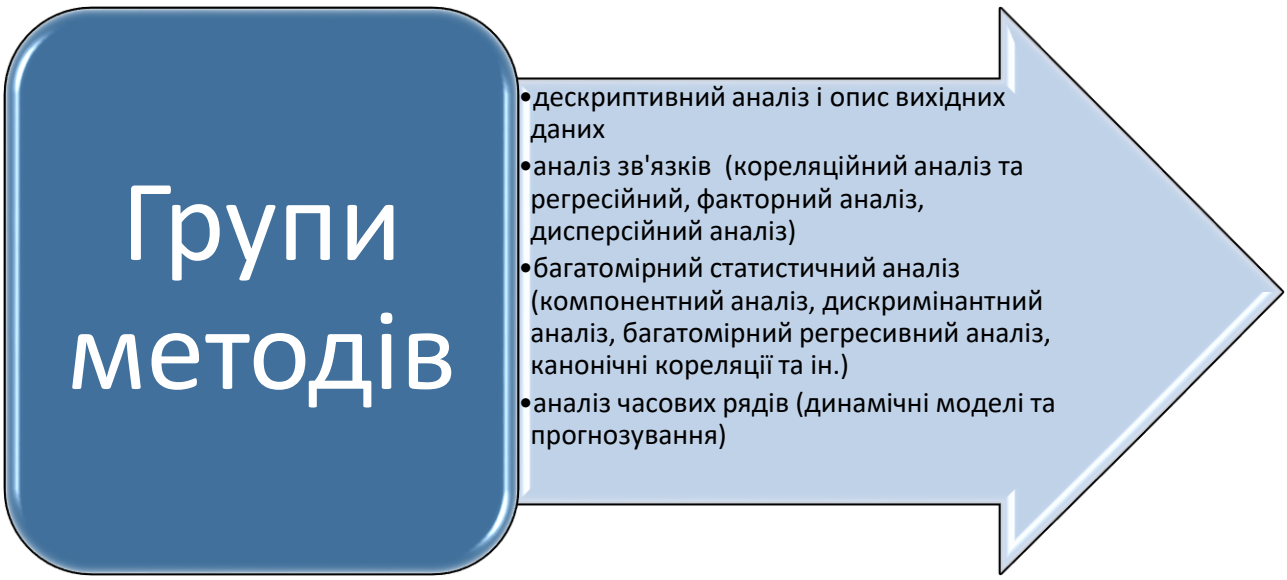


Рисунок 1.2 – Методи Data Mining

Ось деякі методи інтелектуального аналізу даних:

• Кластеризація – або пошук і об'єднання схожих структур і об'єктів. Слово «кластер» в перекладі означає скупчення або гроно. Кластеризація не допомагає робити висновки, а тільки знаходить і об'єднує об'єкти із загальними властивостями.

• Ще одним популярним методом є Алгоритм k-середніх (k-means) (або швидкий кластерний аналіз). Алгоритм k-середніх допомагає визначити гіпотези щодо кількості кластерів. При цьому значення k може залежати від раніше проведених досліджень, припущень або навіть інтуїції.

Рисунок 1.3 – Методи Data Mining

• Ще один популярний метод - байєсовські мережі: графічні структури, які представляють ймовірні стосунки між величезним масивом змінних. Байєсовські мережі служать для створення імовірнісного виведення на основі цих змінних.

• І нарешті, штучні нейронні мережі. Дуже популярна тема останнім часом – і вони у всіх на слуху. Перш ніж скористатися нейронною мережею, її потрібно «навчити». Від того, наскільки правильно, вірно і точно буде навчена мережа, залежить її ефективність у вирішенні тих чи інших завдань. Навчає мережу – людина, аналітик. Тому грамотні фахівці з навчання нейронних мереж дуже затребувані на ринку.

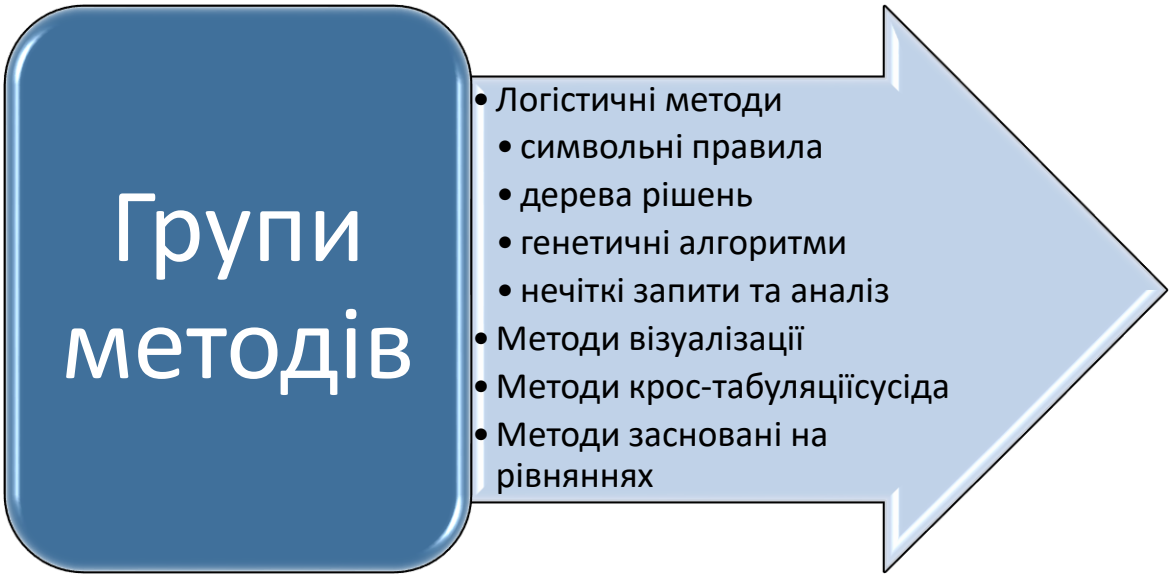


Рисунок 1.4 – Методи Data Mining

Властивості методів Data Mining

Різні методи Data Mining характеризуються певними властивостями, які можуть бути визначальними при виборі методу аналізу даних. Методи можна порівнювати між собою, оцінюючи характеристики їх властивостей.

Основні властивості і характеристики методів Data Mining: точність, масштабованість, інтерпретованість, перевірка, трудомісткість, гнучкість, швидкість і популярність.

У таблиці 1.1 наведено порівняльну характеристику деяких поширених методів. Оцінка кожної з характеристик проведена наступними категоріями, в порядку зростання: надзвичайно низька, дуже низька, низька / нейтральна, нейтральна / низька, нейтральна, нейтральна / висока, висока, дуже висока.

Алгоритм	Точність	Масштабованість	Інтерпретованість	Пригодність	Алгоритм
Лінійна регресія	Нейтральна	Висока	Висока/нейтральна	Висока	Лінійна регресія
Нейронні мережі	Висока	Низька	Низька	Низька	Нейронні мережі
Методи візуалізації	Висока	Дуже низька	Висока	Висока	Методи візуалізації
Дерева рішень	Низька	Висока	Висока	Висока/нейтральна	Дерева рішень
Нейронні мережі	Висока	Нейтральна	Низька	Висока/нейтральна	Нейронні мережі
k-найближчого сусіда	Низька	Дуже низька	Висока/ нейтральна	Нейтральна	k-найближчого сусіда
Алгоритм	Трудомісткість	Різносторонність	Швидкість	Популярність	Алгоритм
Лінійна регресія	Нейтральна	Нейтральна	Висока	Низька	Лінійна регресія
Нейронні мережі	Нейтральна	Низька	Дуже низька	Низька	Нейронні мережі

Методи візуалізації	Дуже висока	Низька	Надзвичайно низька	Висока/нейтральна	Методи візуалізації
Дерева рішень	Висока	Висока	Висока/нейтральна	Висока/нейтральна	Дерева рішень
Нейронні мережі	Низька/нейтральна	Нейтральна	Низька/нейтральна	Нейтральна	Нейронні мережі
k-найближчого сусіда	Нейтральна/низька	Низька	Висока	Низька	k-найближчого сусіда
Алгоритм	Трудомісткість	Різносторонність	Швидкість	Популярність	Алгоритм

Таблиця 1.1 – Порівняльна характеристика деяких поширених методів Data Mining

На сьогодні відомий інтелектуальний аналіз даних, текстовий аналіз звітів і даних і візуальний аналіз даних.

Кожен знає, наскільки легко сприймається візуальний контент – на відміну, навіть від текстового. Не кажучи вже про складний табличний і цифровий. Тому застосування технології візуального аналізу даних необхідно і потрібно там, де такий аналіз можливий.

StatSoft визначає поняття *"Data Mining"* як процес аналітичного дослідження великих масивів інформації (зазвичай економічного характеру) з метою виявлення певних закономірностей і систематичних взаємозв'язків між змінними, які потім можна застосувати до нових сукупностей даних. Цей процес включає три основних етапи: дослідження, побудова моделі або структури і її перевірку. В ідеальному випадку, при достатній кількості даних можна організувати ітеративну процедуру для побудови стійкої (робастної) моделі. У той же час, в реальній ситуації практично неможливо перевірити економічну модель на стадії аналізу і тому початкові результати мають характер евристик, які можна використовувати в процесі прийняття рішення.

Методи *Data Mining* набувають все більшої популярності в якості інструменту для аналізу інформації, особливо в тих випадках, коли передбачається, що з наявних даних можна буде витягти знання для прийняття рішень в умовах невизначеності. Хоча останнім часом зріс інтерес до розробки нових методів аналізу даних, спеціально призначених для сфери інформатизації, в цілому системи *Data Mining* і раніше ґрунтуються на класичних принципах *розвідувального аналізу даних* (РАД) і *побудови моделей* і використовують ті ж підходи і методи.

Є, однак, важлива відмінність процедури *Data Mining* від класичного *розвідувального аналізу даних (РАД)*: системи *Data Mining* більшою мірою орієнтовані на практичне застосування отриманих результатів, ніж на з'ясування природи явища. Іншими словами, при *Data Mining* користувача не дуже цікавить конкретний вид залежностей між змінними задачі. З'ясування природи беруть участь функцій або конкретної форми інтерактивних багатовимірних залежностей між змінними, що не є головною метою цієї процедури. Основна увага приділяється пошуку рішень, на основі яких можна було б будувати достовірні прогнози. Таким чином, в області *Data Mining* прийнятий такий підхід до аналізу даних і вилучення знань, який іноді характеризують словами "чорний ящик". При цьому використовуються не тільки класичні прийоми *розвідувального аналізу даних*, а й такі методи, як *нейронні мережі*, які дозволяють будувати достовірні прогнози, не уточнюючи конкретний вид тих залежностей, на яких такий прогноз заснований.

Дуже часто *Data Mining* трактується як *"суміш статистики, методів штучного інтелекту (II) і аналізу баз даних"* (*Pregibon, 1997, p. 8*), і до останнього часу вона не визнавалася повноцінної областю інтересу для фахівців за статистикою, а часом її навіть називали *"задвірками статистики"* (*Pregibon, 1997, p. 8*). Однак, завдяки своїй великій практичній значущості, ця проблематика нині інтенсивно розробляється і привертає великий інтерес (в тому числі і в її статистичних аспектах), і в ній досягнуті важливі теоретичні результати.

Data Mining часто розглядається як природний розвиток концепції *сховищ даних*.

StatSoft визначає поняття *сховища даних* як спосіб зберігання великих багатовимірних масивів даних, який дозволяє легко витягувати і використовувати інформацію в процедурах аналізу.

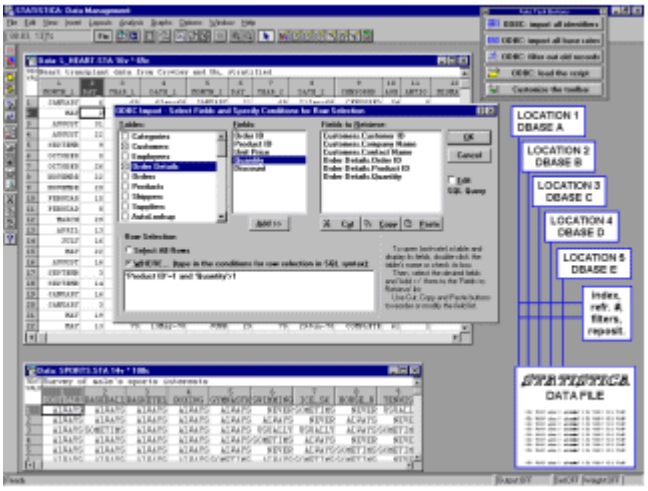


Рисунок 1.5 – *SENS [STATISTICA Enterprise System]* і *SEWSS [STATISTICA Enterprise-Wide SPC System]*

Ефективна архітектура сховища даних повинна бути організована таким чином, щоб бути складовою частиною інформаційної системи управління (або принаймні мати зв'язок з усіма доступними даними). При цьому необхідно використовувати спеціальні технології роботи з корпоративними базами даних (наприклад, *Oracle, Sybase, MS SQL Server).* Високопродуктивна технологія сховищ даних, що дозволяє користувачам організувати і ефективно використовувати базу даних підприємства практично необмеженої складності, розроблена компанією StatSoftenterprise systems і називається *SENS [STATISTICA Enterprise System]* і *SEWSS [STATISTICA Enterprise-Wide SPC System]).*

Оперативна аналітична обробка даних (OLAP)

Термін *OLAP* (або *FASMI* - швидкий аналіз розподіленої багатовимірної інформації) позначає методи, які дають можливість користувачам багатовимірних баз даних в реальному часі генерувати описові та порівняльні зведення ("views") даних і отримувати відповіді на різні інші аналітичні запити. Зверніть увагу, що незважаючи на свою назву, цей метод не має на увазі інтерактивну обробку даних (в режимі реального часу); він означає процес аналізу багатовимірних баз даних (які,

зокрема, можуть містити і динамічно оновлювану інформацію) шляхом складання ефективних "багатовимірних" запитів до даних різних типів. Засоби *OLAP* можуть бути вбудовані в корпоративні (масштабу підприємства) системи баз даних і дозволяють аналітикам і менеджерам стежити за ходом і результативністю свого бізнесу або ринку в цілому (наприклад, за різними сторонами виробничого процесу або кількістю і категоріями укладених угод по різних регіонах). Аналіз, проведений методами *OLAP* може бути як простим (наприклад, таблиці частот, описові статистики, прості таблиці), так і досить складним (наприклад, він може включати сезонні поправки, видалення викидів та інші способи очищення даних). Хоча методи *Data Mining* можна застосовувати до будь-якої, попередньо необробленої і навіть неструктурованої інформації, їх можна також використовувати для аналізу даних і звітів, отриманих засобами *OLAP,* з метою більш поглибленого дослідження. У цьому сенсі методи *Data Mining* можна розглядати як альтернативний аналітичний підхід (службовець іншим цілям, ніж *OLAP)* або як аналітичне розширення систем *OLAP.*

Розвідувальний аналіз даних (РАД)

РАД і перевірка гіпотез

На відміну від традиційної перевірки гіпотез, призначеної для перевірки апріорних припущень, що стосуються зв'язків між змінними, розвідувальний аналіз даних (РАД) застосовується для знаходження зв'язків між змінними в ситуаціях , коли відсутні (або недостатні) апріорні уявлення про природу цих зв'язків. Як правило, при розвідувальному аналізі враховується і порівнюється велике число змінних, а для пошуку закономірностей використовуються найрізноманітніші методи.

Обчислювальні методи РАД

Обчислювальні методи розвідувального аналізу даних включають основні статистичні методи, а також більш складні, спеціально розроблені методи багатовимірного аналізу, призначені для відшукання закономірностей в багатовимірних даних.

Основні методи розвідувального статистичного аналізу. До основних методів розвідувального статистичного аналізу відноситься процедура аналізу розподілів змінних (наприклад, щоб виявити змінні з несиметричним розподілом), перегляд кореляційних матриць з метою пошуку коефіцієнтів, що перевершують по величині певні порогові значення, або аналіз багатовхідних таблиць частот (наприклад, "пошаровий" послідовний перегляд комбінацій рівнів керуючих змінних).

Рисунок 1.6 – Розвідувальний аналіз даних

Методи багатовимірного розвідувального аналізу. Методи багатовимірного розвідувального аналізу спеціально розроблені для пошуку закономірностей в багатовимірних даних (або послідовності одновимірних даних). До них відносяться: кластерний аналіз, факторний аналіз, аналіз дискримінантних функцій, багатовимірне шкалювання, логлінійний аналіз, канонічні кореляції, покрокова лінійна і нелінійна (наприклад, логіт) регресія, аналіз відповідностей, аналіз часових рядів і дерева класифікації.

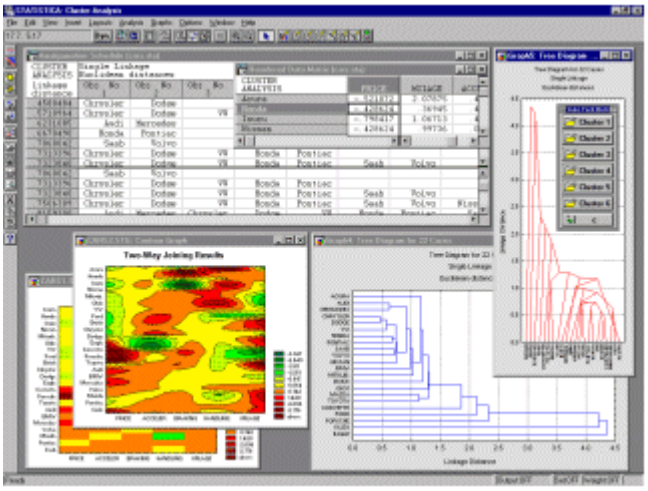


Рисунок 1.7 – Багатовимірний розвідувальний аналіз даних

Нейронні мережі. Цей клас аналітичних методів заснований на ідеї відтворення процесів навчання мислячих істот (як вони представляються дослідникам) і функцій нервових клітин. Нейронні мережі можуть прогнозувати майбутні значення змінних за тим самим які значень цих же або інших змінних, попередньо здійснивши процес так званого *навчання* на основі наявних даних.

Графічні методи РАД (візуалізація даних)

Широкий набір потужних методів розвідувального аналізу даних представлений також засобами графічної візуалізації даних. З їх допомогою можна знаходити залежності, тренди і зміщення, "приховані" в неструктурованих наборах даних.

Зафарбовування. Можливо, найпоширенішим і історично перший з методів, які з повною підставою можна віднести до графічного розвідувального аналізу даних, стало зафарбовування – інтерактивний метод, що дозволяє користувачеві вибирати на екрані комп'ютера окремі точки-спостереження або групи таких точок, знаходити їх характеристики (в тому числі загальні) і вивчати вплив окремих спостережень на співвідношення між різними змінними. Ці співвідношення між змінними також можуть бути візуалізовані за допомогою підгінних функцій разом з відповідними довірчими інтервалами, і, таким чином, користувач може в інтерактивному режимі досліджувати зміни параметрів цих функцій, тимчасово видаляючи або додаючи фрагменти набору даних. За допомогою зафарбовування,

наприклад, можна вибрати (виділити) на одній з матричних діаграм розсіювання всі крапки даних, що належать певній категорії.

Перевірка результатів РАД

Попереднє дослідження даних може служити лише першим етапом у процесі їх аналізу, і поки результати не підтверджені (методами крос-перевірки) на інших фрагментах бази даних або на незалежній безлічі даних, їх можна сприймати найбільше як гіпотезу. Якщо результати розвідувального аналізу свідчать на користь певної моделі, то її правильність можна потім перевірити, застосувавши її до нових даних і визначивши ступінь її узгодженості з даними (перевірка "здатності до прогнозування"). Для швидкого виділення різних підмножин даних (наприклад, для очищення, перевірки та ін.) І оцінки надійності результатів зручно користуватися умовами вибору спостережень.

Нейронні мережі - це клас аналітичних методів, побудованих на (гіпотетичних) принципах навчання мислячих істот і функціонування мозку і дозволяють прогнозувати значення деяких змінних в нових спостереженнях за даними інших спостережень (для цих же або інших змінних) після проходження етапу так званого навчання на наявних даних . Нейронні мережі є одним з методів *Data Mining*.

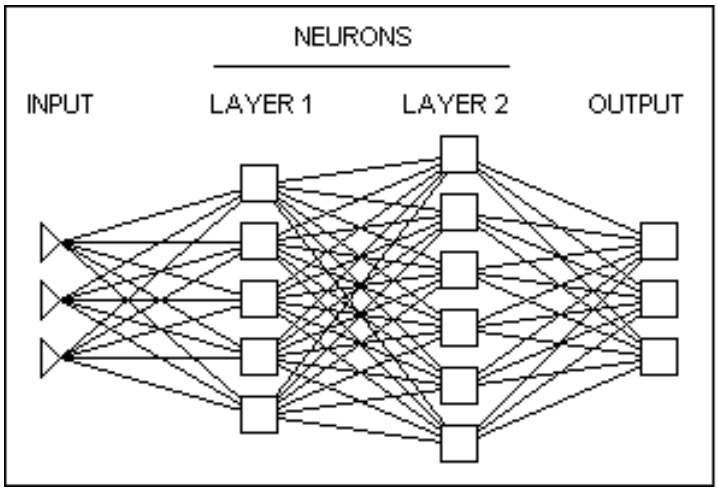


Рисунок 1.8 – Нейронні мережі

При застосуванні цих методів перш за все постає питання вибору конкретної архітектури мережі (числа "шарів" і кількості "нейронів" в кожному з них). Розмір і структура мережі повинні відповідати (наприклад, в сенсі формальної обчислювальної складності) суті досліджуваного явища. Оскільки на початковому етапі аналізу природа явища зазвичай не буває добре відома, вибір архітектури є непростим завданням і часто пов'язаний з тривалим процесом "проб і помилок" (проте, останнім часом стали з'являтися нейронно-мережеві програми, в яких для вирішення цієї трудомісткої завдання пошуку "найкращою" архітектури мережі застосовуються методи штучного інтелекту).

Потім побудована мережа піддається процесу так званого "навчання". На цьому етапі нейрони мережі ітеративно обробляють вхідні даних і коректують свої ваги таким чином, щоб мережа найкращим чином прогнозувала (в традиційних термінах варто було б сказати "здійснювала підгонку") дані, на яких виконується "навчання". Після навчання на наявних даних мережа готова до роботи і може використовуватися для побудови прогнозів.

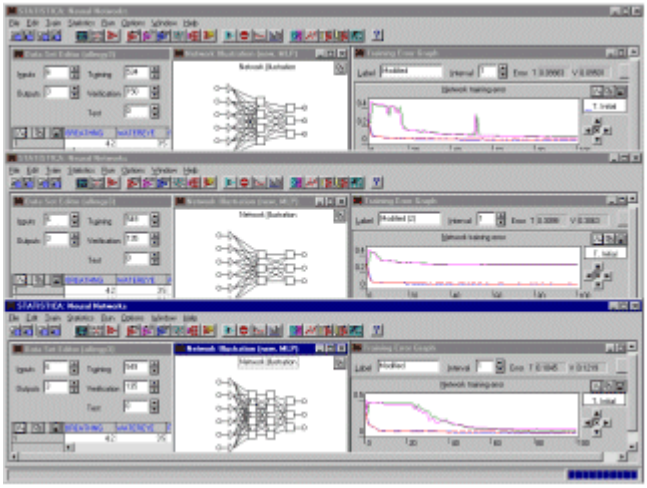


Рисунок 1.9 – Аналіз даних за допомогою нейронної мережі

"Мережа", отримана в результаті "навчання", висловлює закономірності, присутні в даних. При такому підході вона виявляється функціональним еквівалентом певної моделі залежностей між змінними, подібних тим, які будуються в традиційному моделюванні. Однак, на відміну від традиційних моделей, в разі "мереж" ці залежності не можуть бути записані в явному вигляді, подібно до того як це робиться в статистиці.

Іноді нейронні мережі видають прогноз дуже високої якості; однак, вони представляють собою типовий приклад нетеоретичного підходу до дослідження (іноді це називають "чорним ящиком"). При такому підході ми зосереджуємося виключно на практичний результат – в даному випадку – на точності прогнозів і їх прикладної цінності, - а не на суті механізмів, що лежать в основі явища, або відповідно отриманих результатів будь-якої наявної "теорії".

Слід, однак, відзначити, що методи нейронних мереж можуть застосовуватися і в таких дослідженнях, де метою є побудова пояснює моделі явища, оскільки нейронні мережі допомагають вивчати дані на предмет пошуку значущих змінних або груп таких змінних, і отримані результати можуть полегшити процес подальшої побудови моделі . Більш того, зараз є нейромережеві програми, які за допомогою складних алгоритмів можуть знаходити найбільш важливі вхідні змінні, що вже безпосередньо допомагає будувати модель.

Одна з головних переваг нейронної мережі полягає в тому, що вона, принаймні теоретично, може апроксимувати будь-яку безперервну функцію, і тому досліднику немає необхідності заздалегідь приймати будь-які гіпотези щодо моделі, і навіть - в ряді випадків - про те, які змінні дійсно важливі. Однак, суттєвим недоліком нейронних мереж є та обставина, що остаточне рішення залежить від початкових установок мережі та, як уже говорилося вище, її практично неможливо "інтерпретувати" в традиційних аналітичних термінах, які зазвичай застосовуються при побудові теорії явища.

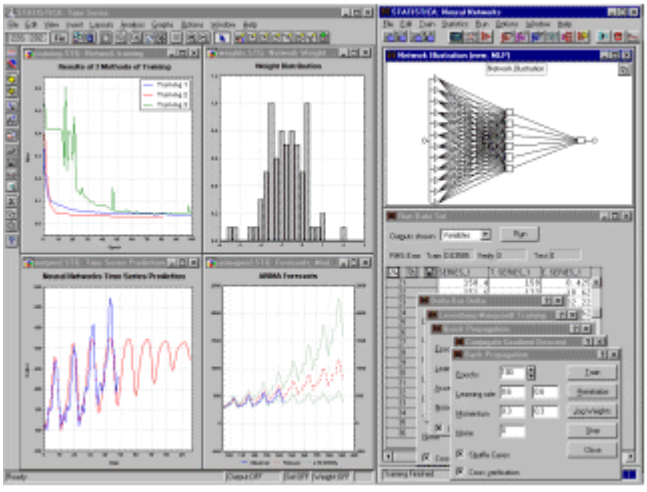


Рисунок 1.10 – Аналіз даних за допомогою нейронної мережі

Деякі автори відзначають той факт, що нейронні мережі використовують або, точніше, припускають використання обчислювальних систем з масовим паралелізмом.

Однак, як зазначає *Ripley (1996),* більшість існуючих нейромережевих програм працюють на однопроцесорних комп'ютерах. На його думку, значне прискорення роботи може бути досягнуто не тільки за рахунок розробки програмного забезпечення, що використовує переваги багатопроцесорних систем, але також шляхом розробки більш ефективних алгоритмів навчання.

1.1. 3. Аналіз даних та атрибути Data mining

У цьому розділі обговорюються різні інструменти інтелектуального аналізу даних з відкритим вихідним кодом для аналізу і побудови предсказательной моделі для класифікації і передбачення без маркування.

WEKA (Waikato Environment for Knowledge Analysis): Університет Вайкато Нової Зеландії розробив цей інструмент з відкритим вихідним кодом в технології Java. Він складається з набору алгоритмів машинного навчання, таких як кластеризація, вибір підбору / вибору атрибутів, класифікація, об'єднання правил і т. Д Weka надає чотири інтерфейси: Explorer, Experimenter, потік знань, простий інтерфейс командного рядка (CLI) для роботи з алгоритмом машинного навчання і наборами даних. Explorer надає платформу для дослідження даних. Експериментатор надає платформу для проведення експериментів для проведення статистичних випробувань серед схем навчання. Потік знань надає графічний інтерфейс користувача для реалізації функцій, доступних в провіднику. SimpleCLI надає простий інтерфейс командного рядка для виконання команд Weka.

Orange: інструмент з відкритим вихідним кодом. Orange містить безліч алгоритмів машинного навчання та інтелектуального аналізу даних з підпрограмами для дослідження даних. Він працює з Python і C ++, він працює для таких функцій, як дерева рішень, вибір підмножини атрибутів, форсування і упаковка. Він дає

платформу для візуального програмування для використання віджетів візуальних компонентів, це досліджує дані для аналізу. Модульність віджета підключає комунікаційний носій для автоматичного обміну пакетами даних для аналізу даних. Orange використовується для багатьох додатків для аналізу даних.

R Tool: Ross Ihaka та Robert Gentleman з Університету Окленда, Нова Зеландія розробили інструмент R в 1996 році. R надає платформу для розрахунку статистики для аналізу даних. , R працює з платформою Unix, Windows, Mac. Формальний робочий потік для завдання інтелектуального аналізу даних виконується за допомогою наступних кроків:

1. Завантажте набір даних і виберіть функції.
2. Вивчіть дані в зрозумілому форматі.
3. Поширення тесту.
4. Перетворіть дані відповідно до моделі.
5. Побудуйте моделі.
6. Оцініть моделі за допомогою набору даних.
7. Перегляньте завдання Log in data mining.

Keen Tool: Keen («Вилучення знань на основі еволюційного навчання»), багато задач з інтелектуального аналізу даних включають в себе алгоритми регресії, контрольованого, неконтрольованого та еволюційного навчання. Він складається з бібліотечних функцій для методу попередньої та пост-обробки даних, для маніпулювання даними, для міфології комп'ютерних обчислень, для сприяння науковим дослідженням в області машинного навчання. Він також підтримує нечіткий і генетичний алгоритм для проведення досліджень в області інтелектуального аналізу даних і різних додатків, пов'язаних з аналізом даних.

1.1.4. Методи, механізми і протоколи мережевої безпеки

Для захисту мережі від загроз і атак є безліч методів, механізмів, пристроїв і протоколів безпеки. Методами безпеки є криптографія, віртуальна приватна мережа (VPN), туннелирование, Хешування, цифровий підпис, центр сертифікації конфігурації вузла Bastion для PKI (інфраструктура відкритого ключа) і т. Д, Механізмами і пристроями захисту є брандмауери, проксі-сервер, демілітаризована зона (DMZ), система виявлення вторгнень, система запобігання вторгнень, сервер доступу до мережі: дистанційна аутентифікація.

(RADIUS), Honey pot, Honey net, Antivirus Software і так далі. Протоколи SSL (захищений рівень сокета) для Secure web, SSH (Secure Shell) [41] для захисту telnet і rlogin або передачі файлів, S / MIME (захищені / багатоцільові розширення електронної пошти Інтернету). Захищена електронна пошта, безпечна інформація Управління журналом управління.

1.1.5. Аналіз системи виявлення вторгнень (IDS)

Вторгнення виникає, коли зловмисник намагається отримати доступ або порушити нормальну роботу мережі. Система виявлення вторгнень вивчає нормальну діяльність мереж і створює прогностичну модель, таку як людська поведінкова модель. На основі цієї моделі він ідентифікує зловмисників в мережі. Методи виявлення вторгнень, що класифіковані як IDS на основі підпису, ідентифікаційні дані, засновані на статистичних аномаліях, аналізатори IDS і протоколів моніторингу стану. Підписи на основі IDS, аналізують мережевий трафік при пошуку шаблонів, які відповідають відомим підписам, тобто попередньо сконфігурованим, визначеним шаблонах атаки. Вузьким місцем цього підходу є те, що новий тип атак має бути ідентифікований і оновлений в базі даних, і це трудомісткий процес. IDS з статистичної аномалією також називають «IDS на основі поведінки». Він збирає статистичні зведення, спостерігаючи за трафіком. Звичайний період оцінки встановлює базовий рівень ефективності. Базові дані можуть

включати такі змінні, як пам'ять хоста або процесор (центральний процесор), типи мережевих пакетів і кількість пакетів. Як тільки базовий рівень встановлений, IDS порівнює мережеву активність з цією базою. Якщо він перевищує базовий рівень, тоді цей рівень відомий як «Рівень відсікання», тоді система IDS негайно відправляє попередження адміністратору. Перевага цього типу полягає в тому, що він може виявляти нові типи атак, оскільки він шукає ненормальну активність будь-якого типу і нестачі, оскільки для цього потрібно набагато більше службових і обчислювальних потужностей, ніж IDS на основі сигнатур. Таким чином, цей метод не підходить для інтенсивного пакетного трафіку.

Робоча група з виявлення вторгнень визначила загальну архітектуру IDS на основі розгляду чотирьох типів функціональних модулів. Ці модулі показані на рисунку 1.11 і є наступними :

1. Модулі подій складаються з сенсорних елементів, контролюючих цільову систему і отримують інформаційні події.

2. Модулі бази даних зберігають інформацію з модулів подій.

3. Модулі аналізу аналізують події і виявляють потенційне ворожа поведінка, при необхідності генеруючи сигнал тривоги.

4. Модулі реагування виконують відповідь, щоб запобігти будь-яке виявлене вторгнення, якщо воно відбувається.

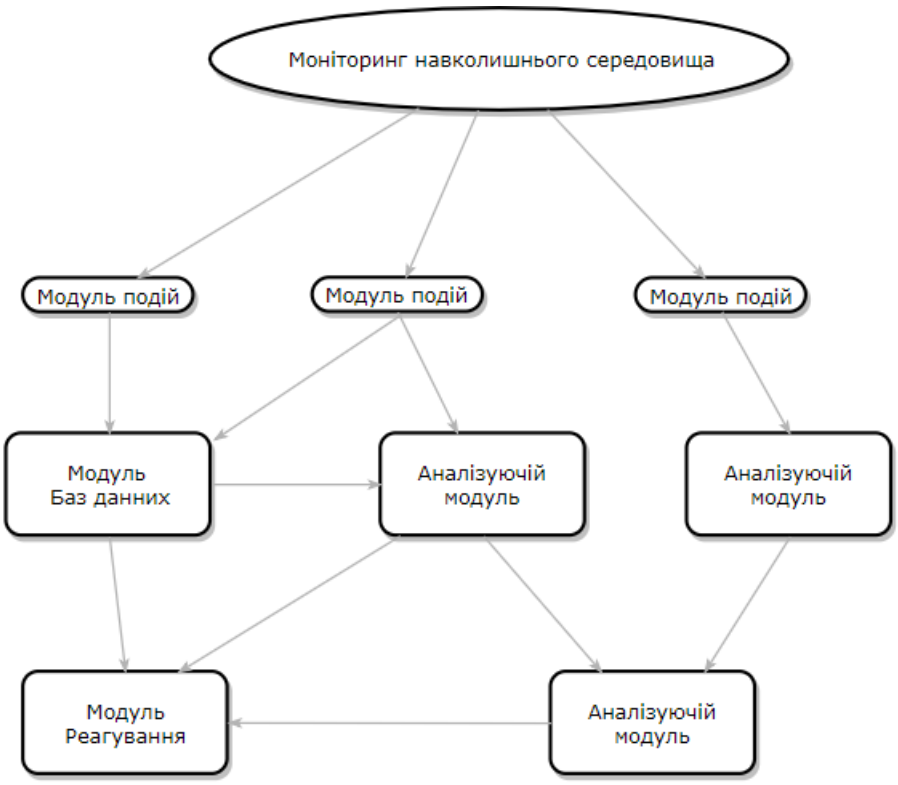


Рисунок 1.11. Загальна архітектура IDS.

Модулі подій отримують переважна кількість даних з контрольованих середовищ. Метою аналізу є обробка даних таким чином, щоб спростити роботу мережевих адміністраторів. Це може бути досягнуто шляхом автоматизації процесів в блоках відповідей або дозволу адміністраторам зосередитися тільки на відповідні події.

Одним з рішень є профілізація мережевого трафіку і інцидентів, записаних в модулях подій. Модуль профілізації як частина модулів аналізу може бути визначений як модуль, який групує подібні мережеві з'єднання і шукає домінуюче поведінку з використанням різних типів алгоритмів. Профілювання зазвичай використовується для розрізнення нормального і аномального мережевого трафіку. Модулі профілювання виконують різні типи алгоритмів або методів для угруповання подібних мережевих підключень, подій або дій і пошуку домінантного поведінки. Робочий процес вікна профілювання показаний на рисунку 1.12 . Він складається з чотирьох кроків:

1. Збір даних
2. Попередня обробка даних
3. Профілізація
4. Звітність

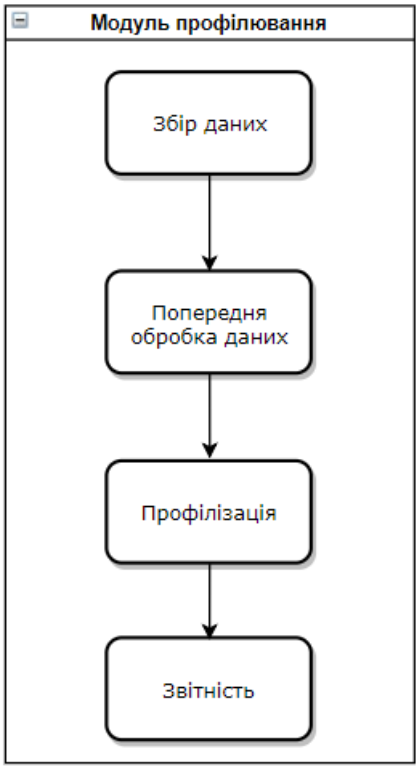


Рисунок 1.12. Робочий процес профілювання в профілюючому модулі.

Дослідники виклали дві з найбільших проблем в профілізації безпеки:

1. Величезний обсяг даних і труднощі у виявленні шаблонів в даних і в вивчених шаблонах

2. Можливість візуалізації, яка може посилити роль профілювання безпеки адміністрацією безпеки.

Метод гібридного виявлення вторгнень був розроблений для підвищення продуктивності і можливостей систем виявлення і запобігання вторгнень (IDPS) шляхом об'єднання методу на основі сигнатур (виявлення неправильного використання) і методу, заснованого на аномалії.

Деякі критерії оцінки, які можуть використовуватися для порівняння продуктивності алгоритмів в IDS, включають в себе:

1) точність, 2) неправдиву негативну швидкість (FNR), 3) неправдиву позитивну швидкість (FPR), 4) використовується час, 5) споживання пам'яті та 6) Статистика Капи. Серед шести критеріїв оцінки для IDS часто використовуються три практичних критерію:

Ÿ Точність: вимірювання відсотку відмови і правильного виявлення, а також кількість помилкових тривог, що генеруються IDS.

Ÿ FNR: відсоток зразків, які повідомляються як нормальні, коли вони фактично аномальні. У ложно-негативної ситуації IDS не може виявити реальних атак.

Ÿ FPR: частка нормальних випадків, які неправильно класифікуються як аномальні.

У таблиці 1.2 показано порівняння характеристик і недоліків між методом аномалії і методом неправильного використання для виявлення вторгнень. У таблиці 1.3 порівнюються три методи виявлення, засновані на різних критеріях ефективності.

Таблиця 1.2. Порівняння методів виявлення

Аспект	Виявлення аномалій	Неправильне використання
Характеристика	Використовує відхилення від звичайних шаблонів використання для виявлення вторгнень, будь-які суттєві відхилення від очікуваної	Використовує шаблони відомих атак (сигнатур) для ідентифікації вторгнень, будь-який збіг з сигнатурами повідомляється як

	поведінки повідомляються як можливі атаки	можлива атака
Недолік	- Помилкові спрацьовування. - Вибір правильного набору системних функцій, що підлягають вимірюванню, є спеціальним і заснованим на досвіді. - Повинен вивчати послідовну взаємозв'язок між транзакціями - Чудові аналітики безпеки	- Помилкові негативи - Неможливо виявити нові атаки - Необхідність поновлення підписів - Відомі атаки повинні бути закодовані вручну - Чудові аналітики безпеки

Таблиця 1.3. Порівняння методів виявлення

Технік и виявле ння	Частот а сигнал ів тривог и	Швидк ість	Спожи вання ресурс ів	Гнучкі сть	Надійн ість	Масшт абован ість	Міцніс ть

Аномальні	Висока	Низька	Висока	Висока	Змінна	Висока	Висока
Підпису	Низька	Висока	Низька	Низька	Висока	Низька	Низька
Гібридні	Змінна	Змінна	Висока	Висока	Висока	Висока	Висока

Дані потоку і його обробка мають наступні характеристики: він динамічний і постійно змінюється, має великий обсяг, допускає тільки одне або невелику кількість сканувань, потоків в і в фіксованому порядку і вимагає швидких відповідей. Потоки потоків веб-трафіку і мережевий трафік є типовими прикладами даних потоку. Ефективний аналіз і управління потоковими даними - величезна проблема, оскільки дані потоку зазвичай не зберігаються в репозиторії даних будь-якого типу. Модель безперервних запитів є типовою моделлю запитів в системі управління потоковими даними, де визначені запити постійно обробляють вхідні потоки, збирають агреговані дані, реагують на зміни потоків даних і повідомляють про свій статус. Стиснення даних потоків включає динамічні зміни і ефективне виявлення загальних шаблонів в даних потоку. Люди зацікавлені в ідентифікації вторгнень, заснованих на аномалії потоку повідомлень, які можуть бути виявлені шляхом динамічного побудови моделей потоків і даних потоку кластеризації або порівняння поточних частих шаблонів з тими, які були в певні попередні часи. Більшість даних потоку знаходяться на низькому рівні абстракції, але аналітикам зазвичай більше цікаві як вищі, так і множинні рівні абстракції. Тому багатовимірний і багаторівневий он-лайн аналіз і видобуток повинні проводитися також по потоковим даними.

Основними проблемами, пов'язаними з інтелектуальним потоком даних, є: еволюція концепції, дрейф концепції і нескінченна довжина. Концептуальна еволюція визначається як розвиток нових класів, в той час як поняття дрейфу означає зміну даних з часом. Нескінченна довжина означає, що дані потоку вимагають нескінченного зберігання і часу навчання. Концептуальний дрейф в

моделі навчання вводиться через компонента швидкості даних великого потоку; зокрема, дрейф концепції вказує, що статистичні властивості цільової змінної, передбаченої моделлю, змінюються з часом непередбачуваним чином. Це серйозна проблема, оскільки передбачення буде менш точним згодом. Виявлення вторгнень в режимі реального часу є трудомістким завданням через великий обсяг даних. Дисбаланс даних також є основною перешкодою. Якщо рівень дисбалансу в даних високий, класифікатори будуть нижче в точності і надійності. Дисбаланс є неминучою проблемою в даних в режимі реального часу через великого розміру і низької частоти деяких транзакцій. Методи вибірки є загальними підходами до зниження впливу дисбалансу на класифікатори.

Шкідливі атаки є динамічними і необхідні для виявлення вторгнень в середовищі реального часу з потоками даних. Подія може бути нормальним саме по собі, але воно зловмисно, якщо воно розглядається як частина послідовності подій. Аналіз даних потоку використовується, щоб допомогти ідентифікувати вторгнення в подібних ситуаціях. Це може бути дуже корисно при ідентифікації послідовностей подій, які часто відбуваються разом, виявлення послідовних патернів і розпізнавання викидів або аномалій. Існують три основні типи аномалій: 1) точкові аномалії - зразки даних, які виявляються як аномалії по відношенню до іншої частини набору даних; 2) колективні аномалії - сукупності вибірок даних, які є аномальними; і 3) контекстуальні (умовні) аномалії - бути аномальними тільки в певних контекстах. Система інформування про ситуацію в реальному часі була розроблена на основі алгоритмів потокової передачі для визначення важливих функцій потоку і виявлення аномального поведінки. Крім того, продуктивність цієї системи була покращена за рахунок удосконалення та покращення добре відомого алгоритму потокової передачі. Ця система використовувалася в живій високошвидкісної мережі середнього масштабу, і були представлені результати роботи і виявлення системи.

Масивний он-лайн аналіз (МОА) є основою для інтелектуального аналізу потоків даних, включаючи інструменти для збору та оцінки алгоритмів машинного

навчання. Він може реалізовувати кластеризацию, класифікацію, регресію, часту розробку графів і часту розробку шаблонів. Він містить онлайн-і автономні колекції для кластеризації та класифікації, а також інструменти оцінки. В даний час MOA є одним з кращих інструментів для інтелектуального аналізу потоків даних. Дані потоку можуть збиратися з різних джерел і оброблятися в процесорі обробки потоку, щоб результати були записані в цільову систему. Flink, Storm і Spark Streaming - це три основні платформи з відкритим вихідним кодом для розподіленої обробки потоків. Flink і Storm в 15 разів вище пропускної здатності, ніж Spark Streaming, мікро-пакетної обробки. Шторм краще справляється з пропускною спроможністю, ніж інші, але Spark Streaming є надійним у відмові вузлів і забезпечує відновлення без втрат. Storm є одним з основних двигунів з розподіленим потоком. Він мав безліч додатків, таких як безперервне обчислення, аналітика в реальному часі, розподілені віддалені виклики процедур (RPC), онлайн-навчання машин і процес ETL (витяг, перетворення і завантаження). Storm є відмовостійким і масштабується, і гарантує, що дані будуть оброблені.

Основною проблемою виявлення аномалій є великий обсяг даних. Методи виявлення аномалій повинні бути ефективними з точки зору обчислювальної ефективності при роботі з входами (даними) у великих розмірах, а входи - це дані потоку, які вимагають оперативного аналізу. Інша проблема - помилкові тривоги через великий обсяг введення. Оскільки вхідні дані можуть містити мільйони об'єктів даних, низький відсоток помилкових тривог може зробити аналіз переважною. Часто доступні марковані дані, відповідні нормальної поведінки, але мітки для вторгнень часто відсутні. Тому часто перевагу надають неконтрольовані або полунаблюдаемие методи виявлення аномалій. Робота з величезними обсягами даних потоку, від структурованих даних до неструктурованих даних, числових даних до текстових потоків в мікро-блогах, є проблемою, оскільки дані потоку є динамічними і можуть бути дуже неоднорідними.

Вибір функцій важливий для машинного навчання та інтелектуального аналізу даних. Видалення надлишкових або нерелевантних функцій і аналіз основних

компонентів (PCA) призводять до зменшення обсягів даних. Вибір функції може поліпшити продуктивність прогнозування моделей за рахунок зменшення розміру даних і прискорення процесу навчання. Вибір функції має безліч додатків, які пов'язані з даними з високим розміром. Більшість досліджень було проведено в рамках автономного підходу до навчання, в якому характеристики навчальних зразків були дані апріорі. Однак таке припущення не підходить в деяких реальних додатках, де навчальні екземпляри можуть надходити в онлайновій формі або дорого збирати всі дані. Було вивчено вибір онлайн-функцій (OFS). Метою вибору онлайн-функцій є розробка онлайн-класифікаторів, які використовують тільки невелику і фіксовану кількість функцій; тому, роблячи онлайн-вибір функцій для видобутку великих даних важливою темою дослідження.

Великі дані визначаються як «набори даних, розмір яких перевищує можливості типових програмних засобів баз даних для захоплення, зберігання, управління і аналізу». Обсяг великих даних у багатьох секторах коливається від декількох десятків терабайт (ТБ: приблизно 1012 байта) до кількох петабайт (PB: приблизно 1015 байт). Великі дані можуть бути занадто великими за обсягом, дуже швидко переміщаються або можуть не відповідати обмеженням звичайних архітектур баз даних. Технології для великих даних включають в себе машинне навчання, інтелектуальний аналіз даних, збір натовпу, обробку природної мови, обробку потоків, аналіз часових рядів, кластерні обчислення, хмарні обчислення, обчислення з паралельним обчисленням, візуалізацію і графічну обробку (GPU) і т. Д. Розподілені файлові системи, файлові системи кластерів і паралельні файлові системи є основними інструментами, використовуваними в великих даних.

1.2. Проблематика дослідження та постановка завдань

1.2.1 Оцінка існуючих вразливостей в сенсорах робототехнічних систем

Оцінка існуючих вразливостей в сенсорах робототехнічних систем базується на дослідженні вразливостей екстероцептивного датчику, камери, мікрофону і т.д.

Загалом роботи володіють цілим рядом вразливостей, серед яких незахищені канали зв'язку, проблеми з аутентифікацією, слабка криптографія і відсутність авторизації.

Дослідникам безпеки вдалося виявити три критичні уразливості. Перша з них полягає в наявності двох прихованих облікових записів для віддаленого підключення. Кожна з них має root-права, паролі для доступу прописані в коді прошивці – в результаті акаунти можна відключити або видалити з системи. У підсумку атакуючий, який знає IP конкретної камери, може легко підключитися до неї по Telnet.

Крім того, зловмисники можуть отримати доступ до адміністративної панелі і зовсім без паролів адміністратора завдяки помилці в роботі системи аутентифікації. Для того, щоб отримати доступ до панелі управління, хакеру потрібно просто перейти за адресою [IP-пристрою]/setup і підібрати параметр "handle" – після цього адміністративна сторінка відкриється без введення пароля. Отримавши до неї доступ, зловмисник може змінювати параметри пристрою і навіть змінювати паролі для всіх користувачів системи.

Третя вразливість призводить до розголошення конфіденційної інформації-проблема виникає через помилку в механізмі обробки користувацьких облікових даних.

1.2.2 Моделювання загроз і проблем безпеки роботів на базі атак

До головних атак, що є можливими у спектрі роботи роботів можна віднести такі атаки, як: Stealth attack, Replay attack, Covert attack, False-Data injection, DoS attack, Remote access, Eavesdropping.

Однією з найбільш вразливих атак є атака Replay attack.

Replay attack є однією з форм мережевої атаки, в якій діє передача даних зловмисно або обманним шляхом або затримкою. Це здійснюється або відправником, або з противником, який перехоплює дані і повторно передає їх,

можливо, як частину *masquerade attack* по *IP packet* заміщення. Це одна з нижчих версій атака " *man-in-the-middle attack* ".

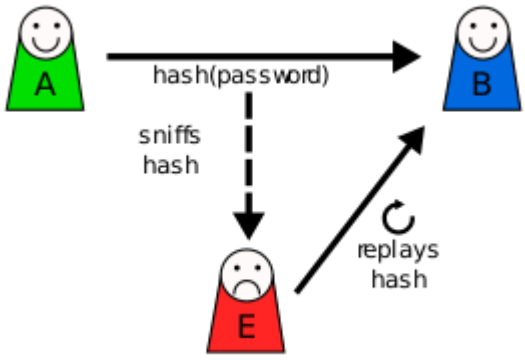


Рисунок 1.13 – Схема реалізації атаки Replay attack

При моделюванні атак необхідно відтворити послідовність дій потенційного порушника – його деструктивні впливи на інформаційну систему при здійсненні атаки. Найбільш поширеними, як зазначається в дослідженні, є моделі атак, засновані на графах (графах атак, байєсовських мережах, мережах Петрі, а також різних розширеннях цих формалізмів). При цьому під графом атак на інформаційну систему розуміється граф, що містить всі відомі траєкторії (сценарії, шляхи) реалізації порушником загроз (цілей). Широко застосовується моделювання атак за допомогою байєсовських мереж. Перевага байєсовських графів атак полягає в тому, що вони представляють собою ймовірні моделі, де переходи між вершинами графа виділяються відповідними умовними ймовірностями, а недоліком є необхідність експертного (в тому числі з використанням різних метрик) завдання ймовірностей виникнення інцидентів, які використовуються порушником при реалізації атаки.

При моделюванні систем захисту інформації використовуються марковські ланцюги або математичний апарат теорії масового обслуговування, але також потрібні експертні завдання такої характеристики безпеки, як ймовірність відбиття атаки системою захисту інформації.

В роботі викладено метод моделювання загрози атаки марковської моделлю з дисконкретними станами і безперервним часом, заснований на введеній в інтерпретації загрози атаки схемою паралельного резервування загроз вразливостей, що створюють її. В результаті моделювання можуть бути розраховані параметри надійності і характеристики загрози атаки.

Шукані параметри названі надійнісними, оскільки моделюється не атакою як процесом послідовного впливу порушником на інформаційну систему, а саме загрозою атаки як процесом виникнення і усунення в системі відмов інформаційної безпеки – реальних загроз атак.

Як випливає з ГОСТ 27.002-2015, надійність – це властивість об'єкта зберігати в часі у встановлених межах значення всіх параметрів, що характеризують здатність виконувати необхідні функції в заданих режимах і умовах застосування, технічного обслуговування, зберігання і транспортування. Виходячи з цього визначення і проводячи моделювання в рамках запропонованої інтерпретації загрози атаки, можна говорити про визначення саме параметрів надійності і характеристик безпеки інформаційної системи, а в загальному випадку – про надійність інформаційної безпеки, під якою розуміємо властивість інформаційної системи з-що зберігаються в часі у встановлених межах значення всіх характеристик безпеки, що визначають здатність системи функціонувати в безпечному режимі. В рамках запропонованого підходу до моделювання завдання захисту інформації інтерпретується як завдання резервування загрозами вразливостей системи захисту загроз вразливостей інформаційної системи, що захищається. Це обумовлюється тим, що для реалізації атаки на захищену інформаційну систему мають бути присутні не тільки всі вразливості інформаційної системи, загрози безпеки яких створюють загрозу атаки, але і всі уразливості системи захисту, що дозволяє говорити про схему паралельного резервування загроз вразливостей.

Перевагою такого підходу до моделювання загрози атаки є можливість об'єктивного (з використанням існуючої статистики, без експертних оцінок)

завдання вхідних параметрів моделі – інтенсивності потоків випадкових подій виникнення та усунення в інформаційній системі загроз вразливостей [9].

У роботах досліджувалися питання побудови марковських моделей надійності інформаційної безпеки – моделей загрози атаки на інформаційну систему.

Загальний підхід до моделювання загрози атаки полягає у приведенні побудованої марковської моделі загрози атаки з дискретними станами і безперервним часом до моделі імовірнісного розріджування вхідних потоків для розрахунку необхідних характеристик загрози атаки. Коректність використання при вирішенні даних завдань моделювання марковських процесів (використовуємо найпростіший потік випадкових подій виникнення в системі вразливостей і експоненціальний розподіл часу усунення вразливостей).

Зауваження. Оскільки моделюється загроза атаки, не потрібно враховувати послідовність використання порушником вразливостей системи. Нехай загроза атаки створюється двома типами загроз вразливостей реалізації (виключені з розгляду загрози технологічних вразливостей, які повинні нівелюватися засобом захисту інформації, розглядаються тільки уразливості, створювані помилками реалізації програмних засобів), з відповідними параметрами – інтенсивністю виявлення і усунення вразливостей (аналогічним чином можна побудувати модель для загрози атаки будь-якої складності).

Стан системи позначимо через, S_{ij}, де i і j – уразливості і- го і j- го типу. Розмічений граф системи станів випадкового (марковського) процесу наведено на рис. 1.14,

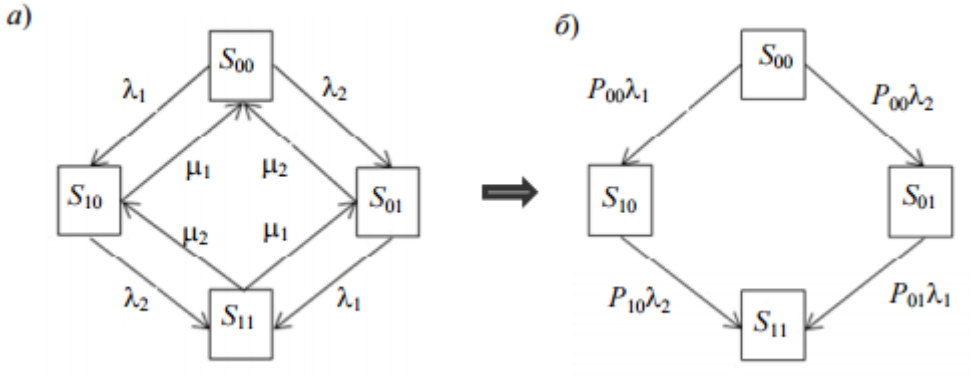


Рисунок 1.14 – Розмічений граф системи станів випадкового (марковського) процесу

Потік подій, що надходять на вхід марковської моделі з інтенсивністю λ, ймовірністно розріджується – розподіляється між станами системи S_{ij}, подія може наступити у випадковий момент часу, коли система знаходиться в одному з можливих станів (переходи між станами у марковській моделі здійснюються миттєво). Розподіл усіх розріджувань найпростішого потоку подій призводить до утворення пронайпростіших потоків подій (рис. 1.14, б).

При побудові цієї моделі виходимо з того, що ймовірність P_{ij} знаходження системи в будь-якому стані у вихідній марковській моделі з дискретними станами та неперервним рівним часом (рис. 1.14, а) інтерпретується як частка часу перебування системи в цьому стані. Принципова відмінність даної моделі, наведеної на рис. 1.14, б, від марковської полягає в тому, що переходи між станами на ній "зважуються" (розмічаються) не інтенсивностями виникнення випадкових подій в системі, а інтенсивностями переходів між станами. Для обґрунтування коректності цього перетворення досить побудувати модель імовірнісного розріджування потоків (всіх потоків, не тільки вхідних) в системі. З використанням моделі імовірнісного розріджування вхідних потоків інтенсивність виникнення в системі реальної загрози атаки може бути розрахована за наступною формулою:

$$\lambda_a = \sum_{S_i \epsilon S_{(R-1)}} P_{S_i} \lambda_{S_i, S_R} \qquad (2.1)$$

де • $S_{(R-1)}$ - безліч станів системи, що характеризуються відсутністю в ній реальної загрози атаки, в кожному з яких система знаходиться з імовірністю $P_{S_{(R-1)}}, S_R$ - стан виникнення в системі реальної загрози атаки. Перехід в стан S_R з $S_{(R-1)}$ в системі здійснюється з інтенсивністю $\lambda_{S_{(R-1)}, S_R}$ Наприклад, для моделі, представленої на рис. 2.2, б:

$$\lambda_a = P_{10}\lambda_2 + P_{01}\lambda_1 \qquad (2.2)$$

Оскільки в стаціонарному режимі функціонування за частку часу перебування системи в стані виникнення реальної загрози атаки ($1 - P_{0A}$ де P_{0A} - ймовірність готовності системи до безпечної експлуатації у відношенні загрози атаки) зі стану, характеризує реальну загрозу атаки, виходить що потік подій λ_a, що надходить в нього (Система без втрат, всі уразливості усуваються), можливо розрахувати інтенсивність усунення з системі реальних загроз атак:

$$\mu_a = \frac{\lambda_a}{1 - P_{0a}} \qquad (2.3)$$

Для розглянутого прикладу

$$\mu_a = \frac{P_{10}\lambda_2 + P_{01}\lambda_1}{P_{11}} \qquad (2.4)$$

Імовірність готовності інформаційної системи до безпечної (щодо загрози атаки) експлуатації можна визначити як

$$P_{0a} = P_{00} + P_{10} + P_{01} = \frac{\mu_1\mu_2 + \lambda_1\mu_2 + \lambda_2\mu_1}{(\lambda_1 + \mu_1)(\lambda_2 + \mu_2)} \qquad (2.5)$$

середній час напрацювання на відмову безпеки інформаційної системи (система, що відновлюється) щодо загрози атаки T_{0ya}, середній час відновлення безпеки інформаційної системи $T_{вуа}$ щодо загрози атаки:

$$T_{\text{вуа}} = \frac{1}{\mu_{\text{а}}}, T_{0\text{уа}} = \frac{1}{\lambda_{\text{а}}} - T_{\text{вуа}} \qquad (2.6)$$

Зауваження. Згідно рис. 1.14, а, $\frac{1}{\lambda_{\text{а}}}$ - середній час напрацювання системи між відмовами безпеки $T_{0\text{уа}}$, $T_{\text{вуа}}$.

Визначимо коректність використання для моделювання загрози атаки кінцевої (з кінцевим числом станів) марковської моделі з дискретними станами і неперервчастим часом. Дослідити дану проблему нам дозволить модель імовірнісного розріджування вхідних потоків.

Зауваження. Якщо число можливих станів звичайно або лічильно (їм можуть бути присвоєні порядкові номери), то випадковий процес називається процесом з дискретними станами.

Визначимо інтенсивність потоку подій, що циркулює в моделі імовірнісного розріджування вхідних потоків, створюваного виникненням і усуненням першої вразливості. На вхід моделі для цієї загрози надходить найпростіший потік подій з інтенсивністю λ_1, переводячи систему зі стану S_{00}, в якому вона знаходиться з імовірністю P_{00}, і із S_{10} з P_{10} (розподіляється між двома станами системи S_{00} і S_{10}). Інтенсивність потоку подій, що циркулює в моделі, визначається наступним чином:

$$\lambda_{n1} = (P_{00} + P_{01})\lambda_1 < \lambda_1 \qquad (2.7)$$

Викликано це протиріччя ($\lambda_{n1} < \lambda_1$) тим, що не з усіх станів марковської моделі є переходи, створювані потоком подій з інтенсивністю λ_1 надходять на вхід марковської моделі, - переходи відсутні для станів S_{10} і S_{11}, вхідний потік розріджується не між усіма станами, тобто в системі присутні інтервали часу, з плином яких події в систему не надходять, що не дозволяє говорити про коректність використання в цьому випадку найпростішого (стаціонарного пуасонівського) вхідного потоку випадкових подій.

Похибка моделювання для цього прикладу тим більше, чим більше $P_{10} + P_{11}$ (В загальному випадку – це сума значень ймовірностей подій, з яких не виходить аналізований потік подій).

Сформулюємо і доведемо кілька важливих тверджень, що стосуються розглянутої проблеми моделювання загрози атаки.

1. Модель загрози атаки як системи без втрат з дискретними станами та неперервним рівним часом коректна в загальному випадку, якщо з кожного стану на графі системи станів випадкового процесу виходять всі I вхідні потоки подій з інтенсивністю $\lambda_i, i = 1, \dots, I$.

Доведення. Тільки при виконанні цієї умови в загальному випадку для всіх I вхідних потоків подій буде виконуватися умова: $\lambda_{ni} = \lambda_i$, що підтверджує коректність імовірнісного розріджування вхідних потоків для цієї моделі і обумовлює можливість визначення на такій моделі параметрів безпеки загрози атаки.

2. У загальному випадку для моделювання загрози атаки повинні використовуватися рахункові (З нескінченним числом станів) марковські моделі з дискретними станами та неперервним рівним часом.

Доведення. Умова, що з кожного стану на графі системи станів випадкового процесу виходять всі I вхідних потоків подій, здійснимо тільки при безкінечному числі станів на графі.

Умовою коректності марковської моделі з дискретними станами і безперервним часом без втрат є коректний імовірнісний розподіл вхідного потоку випадкових подій між можливими станами системи.

Така модель є лічильною.

Така модель може застосовуватися для математичного моделювання об'єктів, які характеризуються можливістю одночасного виникнення в системі двох і більше випадкових подій одного типу.

Така модель може застосовуватися для математичного моделювання загроз атак, оскільки в системі можливе одночасне виникнення кількох загроз вразливостей реалізації одного типу.

Коректна марківська модель загрози атаки для рис. 1.14, а представлена на рис. 1.15.

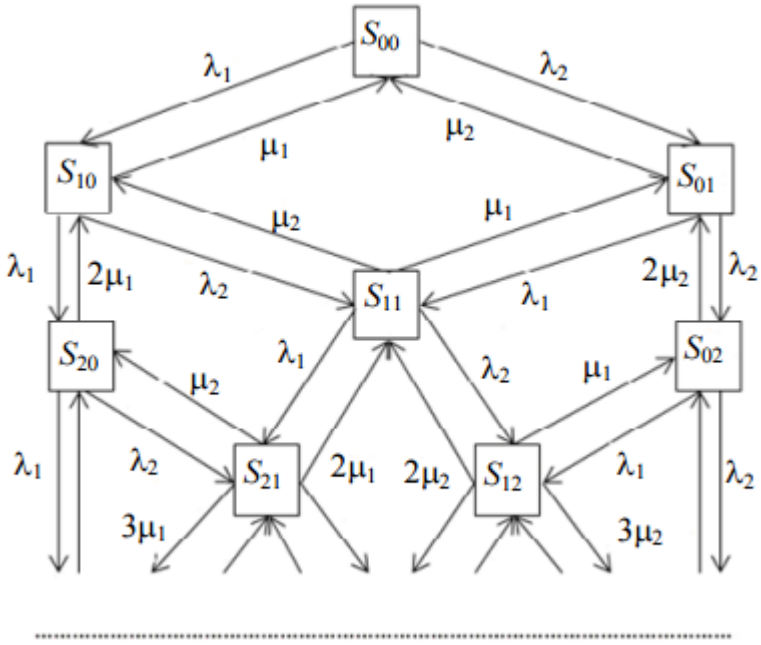


Рисунок 1.15 – Коректна марківська модель загрози атаки

Оскільки для розрахунку параметрів надійності і характеристик загрози атаки необхідна модель з кінцевим числом станів (моделюються переходи між станами системи), визначимо, яким чином можна перейти до подібної моделі за допомогою введення при моделюванні відповідних припущень. Оскільки при моделюванні використовується найпростіший потік, скористаємося законом Пуассона [10]. Нас цікавить ймовірність виникнення в системі одночасно (не одномоментно) кількох подій – одночасне виявлення декількох вразливостей одного типу на інтервалі часу усунення вразливостей цього типу, середньої тривалості $t = 1 / \mu$. Використовуючи коефіцієнт навантаження $\rho = \lambda / \mu$, визначимо необхідну ймовірність одночасної появи в системі m подій:

$$P_m(\rho) = \frac{\rho^m}{m!} e^{-\rho} \qquad (2.8)$$

Точність моделі залежить від того, стаціонарної ймовірністю яких станів знехтувано. Для кожного типу загрози вразливостей, з урахуванням заданих вимог до точності моделювання, за допомогою розрахунку значень ймовірностей $P_m(\rho)$

визначається число max_i враховуються при моделюванні вразливості одного типу, що одночасно виникають в системі.

Всі стани $S_{i>\max ij}$ і дуги між ними виключаються з розміченого графа системи станів випадкового процесу лічильної моделі, в результаті чого виходить шукана кінцева модель загрози атаки, коректність (в частині, що вводяться припущень) якої обґрунтовується тим, що ймовірність події – поява одночасно *max i + 1* вразливостей одного типу - не позначається на результатах моделювання.

У модель загрози атаки захищеної інформаційної системи з використанням в ній системи захисту інформації (СЗІ) в гарячому резерві включаються загрози уразливості СЗІ, що виступають в якості резервують елементів.

Нехай загроза атаки на інформаційну систему (рис. 1.16, б) створюється однією загрозою вразливостей реалізації, стосовно до моделювання якої приймемо допущення про те, що ймовірністю появи в системі одночасно більше двох типів вразливостей можна знехтувати, і загроза атаки на СЗІ також створюється загрозою вразливостей одного типу (рис. 1.16, а; ймовірністю одночасної появи більше однієї подібної уразливості можна знехтувати).

Для таких припущень граф системи станів випадкового процесу для загрози атаки на захищену інформаційну систему наведено на рис. 1.16, в, де і - число виявлених вразливостей інформаційної системи, j - число виявлених вразливостей СЗІ, $\lambda_{\text{a}}, \mu_{\text{a}}$ - параметри безпеки загрози атаки на інформаційну систему, $\lambda_{\text{СЗІ}}, \mu_{\text{СЗІ}}$ - параметри безпеки загрози атаки на СЗІ.

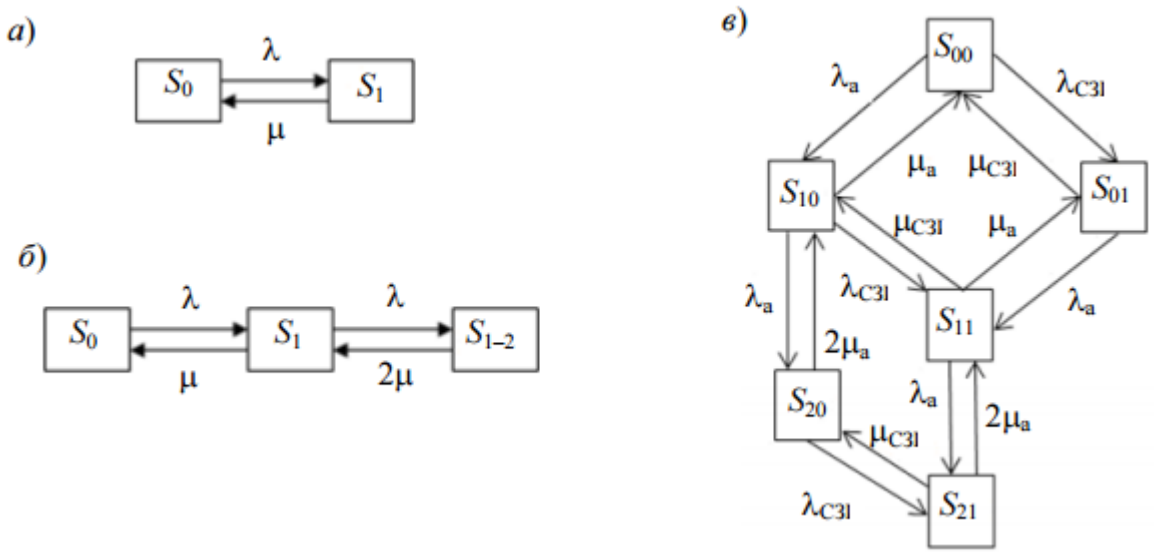


Рисунок 1.16 – Граф системи станів випадкового процесу для загрози атаки на захищену інформаційну систему

Використовуючи таку коректну марковську модель загрози атаки, можна визначити параметри надійності і характеристики безпеки загрози атаки на захищену інформаційну систему, що передбачає побудову моделі імовірнісного розріджування вхідних потоків. Оскільки в системі одночасно можуть бути присутніми одна (стан S_{11}) або дві реальних загрози атаки (S_{21}), визначимо для цих випадків відповідно:

$$\lambda_{a1} = P_{10}\lambda_{\text{СЗІ}} + P_{01}\lambda_a \qquad (2.9)$$

$$\lambda_{a2} = P_{20}\lambda_{\text{СЗІ}} + P_{11}\lambda_a \qquad (2.10)$$

Зауваження. Розраховуючи інтенсивність λ_{a1} можна оперувати тільки вхідним потоком подій, оскільки саме він визначає виникнення в системі реальних загроз атак внаслідок вразливостей реалізації. Зокрема, при цьому не розглядається потік подій з інтенсивністю $P_{21}2_{\mu_a}$, що переводить систему зі стану S_{21} в S_{11} (рис. 1.16), оскільки в цьому випадку нова уразливість в системі не виникає, а усувається одна з вразливостей, що виникла в системі.

Відповідно μ_{a1} і μ_{a2} визначаються наступним чином:

$$\mu_{a1} = \frac{P_{10}\lambda_{\text{СЗІ}} + P_{01}\lambda_a}{P_{11}} \qquad (2.11)$$

$$\mu_{a2} = \frac{P_{20}\lambda_{\text{СЗІ}} + P_{11}\lambda_a}{P_{21}} \qquad (2.12)$$

Істотно розширити можливості моделювання можна, об'єднавши стани в моделі імовірнісного розріджування вхідних потоків випадкових подій (що дозволяє використання при моделюванні найпростішого вхідного потоку).

Для розрахунку параметрів відмов і відновлень безпеки щодо загрози атаки λ_0 і $\mu_{\text{в}}$ (де під відмовою безпеки будемо розуміти виникнення в системі хоча б однієї реальної загрози атаки, під відновленням безпеки – усунення всіх реальних загроз атак, що виникають) в моделі імовірнісного розріджування вхідних потоків вимагається об'єднати їхні капітали системи, що характеризують наявність хоча б однієї реальної загрози атаки зі збереженням всіх вихідних переходів в (з) отриманий подібним чином стан. Представлені на рис. 1.16 стани S_{11} або S_{21} потрібно об'єднати в стан S_2 (при цьому $P_2 = P_{11} + P_{21}$):

$$\lambda_0 = \lambda_{\text{СЗІ}}(P_{10} + P_{20}) + P_{01}\lambda_a \qquad (2.12)$$

З урахуванням того, що безпека системи, порушувана з інтенсивністю λ_0 встановлюється за частку часу $P_2 = P_{11} + P_{21}$ інтенсивність відновлення розраховується по формулою:

$$\mu_{\text{в}} = \frac{\lambda_{\text{СЗІ}}(P_{10}+P_{20})+P_{01}\lambda_a}{P_2} = \frac{\lambda_{\text{СЗІ}}(P_{10}+P_{20})+P_{01}\lambda_a}{P_{11}+P_{21}} \qquad (2.13)$$

Відповідним чином розраховуються тимчасові характеристики безпеки щодо загрози атаки – середній час між відмовами безпеки інформаційної системи щодо загрози атаки $T_{м0о}$, середній час напрацювання на відмову безпеки інформаційної системи (відновлювана система) щодо загрози атаки $T_{0о}$, середній час відновлення безпеки інформаційної системи $T_{0в}$:

$$T_{\text{м0о}} = \frac{1}{\lambda_o};\ T_{0\text{в}} = \frac{1}{\mu_{\text{в}}};\ T_{0o} = T_{\text{м0о}} - T_{0\text{в}} \qquad (2.14)$$

Імовірність готовності інформаційної системи до безпечної експлуатації у відношенні загрози атаки (рис. 2.4) визначається наступним чином:

$$P_{0a} = P_{00} + P_{10} + P_{01} + P_{20} \qquad (2.15)$$

Коректне об'єднання станів на графі системи станів випадкового процесу відбувається, якщо з об'єднаних станів під впливом одного і того ж потоку випадкових подій реалізуються переходи в один і той же, в тому числі об'єднаний стан. Під холодним резервуванням загроз вразливостей реалізації СЗІ будемо розуміти такий режим її експлуатації, при якому загроза умовної технологічної вразливості починає нівелюватися після того, як стане відомо про виникнення відповідної уразливості, і триває до її усунення. Використання СЗІ в холодному резерві дозволяє знизити навантаження на обчислювальні ресурси.

Таким чином, холодне резервування тут можна розглядати як якийсь компроміс між продуктивністю (вплив СЗІ на завантаження обчислювальних ресурсів) і безпекою інформаційної системи: захист використовується тільки при реальній загрозі атаки, причому після того, як про її виникнення стає відомо.

В інформаційній безпеці термін "zero-day" позначає вразливість, яка відома ("опублікована") і не усунута в системі, як наслідок, може використуватися при реалізації атаки на інформаційну систему. Саме "опублікування" вразливостей реалізації дозволяє ініціювати процес нейтралізації створюваних ними загроз технологічних вразливостей СЗІ.

Є певний проміжок часу, протягом якого про виниклу в системі вразливість знає тільки потенційний порушник: з моменту виявлення уразливості до її "опублікування". Протягом цього часу при холодному резервуванні загроз вразливостей потенціальний порушник може атакувати інформаційну систему, оскільки ще немає підстав для початку нейтралізації відповідної загрози умовної технологічної вразливості.

РОЗДІЛ 2 АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ НА БАЗІ DATA MINING

2.1 Дослідження існуючої системи дослідження SIEM по обробці великої кількості даних

2.1.1 Загальний опис системи SIEM

Основоположним принципом системи SIEM є те, що релевантні дані про безпеку підприємства виробляються в різних місцях, і можливість перегляду всіх даних з однієї точки зору полегшує виявлення тенденцій і розглядає шаблони, які незвичайні, SIEM об'єднує SIM (управління інформаційною безпекою) і SEM (управління подіями безпеки) в одну систему управління безпекою. Система SIEM збирає журнали та іншу документацію, пов'язану з безпекою, для аналізу. Більшість систем SIEM працюють шляхом розгортання декількох агентів збору в ієрархічному порядку для збору пов'язаних з безпекою подій від пристроїв, серверів, мережевого обладнання кінцевого користувача і навіть спеціалізованого обладнання безпеки, такого як брандмауери, антивірусні системи або системи запобігання вторгнень. Колектори пересилають події на централізовану консоль управління, яка виконує перевірки і аномалії прапорів. Щоб система могла ідентифікувати аномальні події, важливо, щоб адміністратор SIEM спочатку створив профіль системи в звичайних умовах події.

За своєю суттю SIEM забезпечує:

1) Збір подій і журналів - це може відбуватися у багатьох формах, особливо у власних додатках.

2) Шаруваті центральні види - Зазвичай це у вигляді панелей інструментів або «уявлень»

3) Нормалізація - функція з двох частин. Це включає в себе переклад комп'ютеризованого жаргону на читаються дані для відображення і зіставлення даних з класифікаціями / характеристиками, визначеними користувачем або постачальником. Це іноді називають «зіставлення полів».

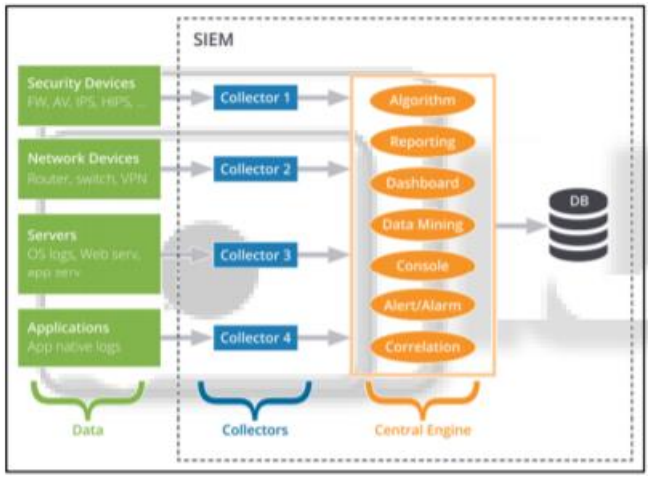
4) Кореляція - це істотно дає контекст даних і формує відносини, засновані на правилах, архітектурі і попередженнях. Це повинно бути як історичним, так і реальним.

5) Адаптивність (масштабованість) - це нерозумно, щоб говорити на будь-якій мові незалежно від джерела, формату, типу, зміни або відповідності вимогам.

6) Звітність та оповіщення - це може бути використано не тільки для того, щоб показати цінність для керівників, а й забезпечити автоматичну перевірку безперервного моніторингу, тенденцій і аудиту. Деякі стверджують, що аудиторський аспект є важливою функцією, але сам SIEM нічого не робить.

7) Управління журналом

Надання можливості для зберігання подій і журналів в центральному місці, а також для забезпечення відповідності вимогам зберігання або зберігання. (Знову ж, багато хто стверджує, що це окрема функція, і я б не погодився.)



Малюнок 2.1. Архітектура SIEM

Архітектура SIEM складається з чотирьох основних частин:

1) Data Sources: система SIEM отримує потік даних з різних пристроїв, які включають не тільки мережеві пристрої, але і деякі фізичні пристрої захисту, такі як біометричні пристрої, зчитувачі карт.

2) Data Collectors: основною функцією збирача даних є нормалізація. Ця нормалізація відбувається двома способами: спочатку нормалізує такі значення, як годинниковий пояс, пріоритет, ступінь важливості в загальному форматі, потім вони нормалізують структуру даних в загальний формат. Деякий час колекціонер робить агрегацію, наприклад, якщо є 5 подібних подій менш ніж за 3 секунди, тоді колекціонер може відправити тільки одна така подія. Ця фільтрація підвищує ефективність і точність і скорочує час обробки.

3) Central Engine: це серце системи SIEM, яке в основному застосовує алгоритм інтелектуального аналізу даних. Цей движок записує події в базу даних у міру їх потоку в систему. Він одночасно обробляє їх через механізм інтелектуального аналізу даних, де відбувається кореляція. Він також має користувальницький інтерфейс для відображення результату алгоритму інтелектуального аналізу даних. Він дозволяє кінцевим користувачам змінювати певні властивості алгоритму. Деякі з інших компонентів цього движка подають звіти, оповіщення та інформаційні панелі.

4) Data Base. У міру того, як потоки подій надходять до центрального движок, вони записуються в базу даних з нормалізованої схемою. Це сховище допомагає нам проводити криміналістичний аналіз історичних даних. Зберігаючи події, ми можемо протестувати новий алгоритм за історичними даними.

2.1.2 Data mining техніки в SIEM

Основна концепція правил асоціації та її використання в SIEM:

В процесі розробки правил спільноти виявляються часті шаблони, асоціації, кореляції або причинні структури серед великих груп предметів або об'єктів в транзакційних базах даних, реляційних базах даних та інших інформаційних сховищах. Короткий виклад концепції правил об'єднання в транзакційних і реляційних базах даних представлено наступним чином: I = {i1, i2, i3, ..., in} - набір елементів, нехай DB - це набір транзакцій бази даних, де кожна транзакція T являє

собою набір таких елементів, що $T \subseteq I$. Кожна транзакція пов'язана з унікальним ідентифікатором транзакції (TID). Нехай X, Y - безліч елементів, правило асоціації - це висновок виду $X \rightarrow Y$, де $X \subset I, Y \subset I$, і X∩Y = ø. X називається антецедентом правила, а Y називається наслідком правила. Набір елементів, що містить і itemset, званий i-itemset. Правило $X \rightarrow Y$ виконується в наборі транзакцій D з підтримкою S, який являє собою відсоток транзакцій, які містять як елементи X і Y, що входять в ту ж транзакцію. Для того, щоб набір предметів був цікавим, його підтримка повинна бути вище, ніж заданий користувачем мінімум. Кажуть, що такі набори предметів є частими.

$$Support(X \rightarrow Y) = P(X \cup Y)$$

Існує ще одна міра Confidence C, де C - відношення кількості транзакцій в D, яке містить X і Y, до числа транзакцій, які містять тільки X як рівняння

$$Confidence(X \rightarrow Y) = P(Y|X) = Support(X \cup Y)/Support(X)$$

Розробка правил управління асоціаціями - це процес пошуку всіх правил асоціації, які передають умову мінімальної підтримки та мінімальної впевненості. Щоб розміновувати ці правила, спочатку значення підтримки і довіри повинні обчислюватися для всіх правил, а потім порівнювати їх із граничними значеннями, щоб обрізати правила з низькими значеннями підтримки або впевненості. У загальному правилі асоціації можна підсумувати в два етапи

1) Знайдіть великі набори елементів, безліч елементів, які підтримують транзакцію вище заданого мінімального порога.

2) Використовуйте великі набори елементів для створення правил асоціації для бази даних, яка має впевненість вище визначеного мінімального порога.

Асоціативні техніки використовуємі в SIEM:

1) Алгоритм Apriori, представлений Agrawal Apriori, є найбільш важливим алгоритмом для пошуку частих наборів елементів для правил логіки Boolean в даній

базі даних. Він використовує властивість: всі непусті підмножини частих наборів предметів також повинні бути частими. Ключовою ідеєю алгоритму Apriori є створення декількох проходів над базою даних. Він використовує ітеративний підхід, відомий як пошук по ширині (пошук по рівню) через пошукове простір, де набори k-елементів використовуються для вивчення (k + 1) - наборів предметів. Спочатку знайдений набір частих 1-предметів. Набір містить один елемент, що задовольняє порогу підтримки, і позначається L1. В кожному наступному проході ми починаємо з набору насіння наборів предметів, які, як було встановлено, є великими в попередньому проході. Цей набір насіння використовується для розробки нових потенційно великих наборів елементів, званих наборами елементів-кандидатів, і для підрахунку фактичної підтримки цих наборів елементів-кандидатів під час проходження по даним. В кінці проходу ми вирішуємо, які з наборів елементів-кандидатів фактично є великими (частими), і вони стають насінням для наступного проходу. Тому L1 використовується для пошуку L2, набору частих наборів елементів 2, які використовуються для пошуку L3 і т. Д., Поки не будуть знайдені більш часті набори k-елементів. Потім для зменшення простору пошуку використовується дуже важлива властивість, зване властивістю Apriori. Очевидно, алгоритм Apriori складається з кроків з'єднання і обрізки.

2) Частотний алгоритм зростання

FP-алгоритм зростання - ефективний метод розробки всіх частих наборів елементів без генерації кандидатів. FP-зростання використовує комбінацію вертикальних і горизонтальних макетів бази даних для зберігання бази даних в основній пам'яті. Замість того, щоб зберігати обкладинку для кожного елемента в базі даних, він зберігає фактичні транзакції з бази даних в деревоподібної структурі, і кожен елемент має пов'язаний список, що проходить через всі транзакції, що містять цей елемент. Ця нова структура даних позначається FP-деревом. По суті, всі транзакції зберігаються в структурі даних дерева. Кожен вузол додатково зберігає лічильник, який відстежує кількість транзакцій, які поділяють гілка через цей вузол. Крім того, зберігається посилання, яка вказує на наступне входження відповідного

елемента в FP-дереві, так що все входження елемента в FP-дереві пов'язані один з одним. Крім того, таблиця заголовків зберігається разом з кожним окремим елементом разом зі своєю підтримкою і посиланням на перше входження елемента в FP-дереві. В FP-дереві всі елементи впорядковані в порядку убування підтримки, так як є надія, що це уявлення бази даних буде якомога менше, оскільки все частіше зустрічаються елементи розташовані ближче до кореня дерева FP і, отже, з більшою ймовірністю будуть розділятися.

РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИКОРИСТОВУЮЧИ ІНСТРУМЕНТИ HADOOP

3.1 Архітектурні тактики Security Analytics Systems

У цьому розділі представлені результати аналізу даних по архітектурної тактиці систем аналітики безпеки. Архітектурна тактика - це стратегія проектування, яка впливає на досягнення атрибута якості. Архітектурна тактика, представлена в цьому розділі, була викликана тим, що розглянуті дослідження, засновані на 1) явно виражених атрибутах якості; 2) атрибутах якості, що виводиться із заявленої архітектури для пропонованої системи, і 3) компоненти і їх взаємозв'язок, в архітектурі пропонованої системи. На рисунку 3.1 показано ідентифікована тактика. Було визначено шість тактик для продуктивності, чотири для точності, два для масштабованості, три для надійності і один для забезпечення безпеки і зручності використання. Ця тактика представлена з використанням наступного шаблону.

• Введення: коротке пояснення того, як тактика досягає бажаного атрибута якості;

• Мотивація: обгрунтування того, чому тактика повинна бути включена в проект архітектури;

• Опис: докладне пояснення того, як різні компоненти системи взаємодіють для досягнення бажаного атрибута якості, і діаграми архітектури, що виділяє компоненти, пов'язані з тактикою;

• Обмеження: необхідні умови для включення тактики в існуючу архітектуру системи;

• Приклад: одна або кілька систем з розглянутих досліджень, що демонструють застосування тактики;

• Залежності: чи залежить ця тактика від іншої тактики (ів) для її включення в систему;

• Зміна (необов'язково): злегка змінена форма вихідної тактики

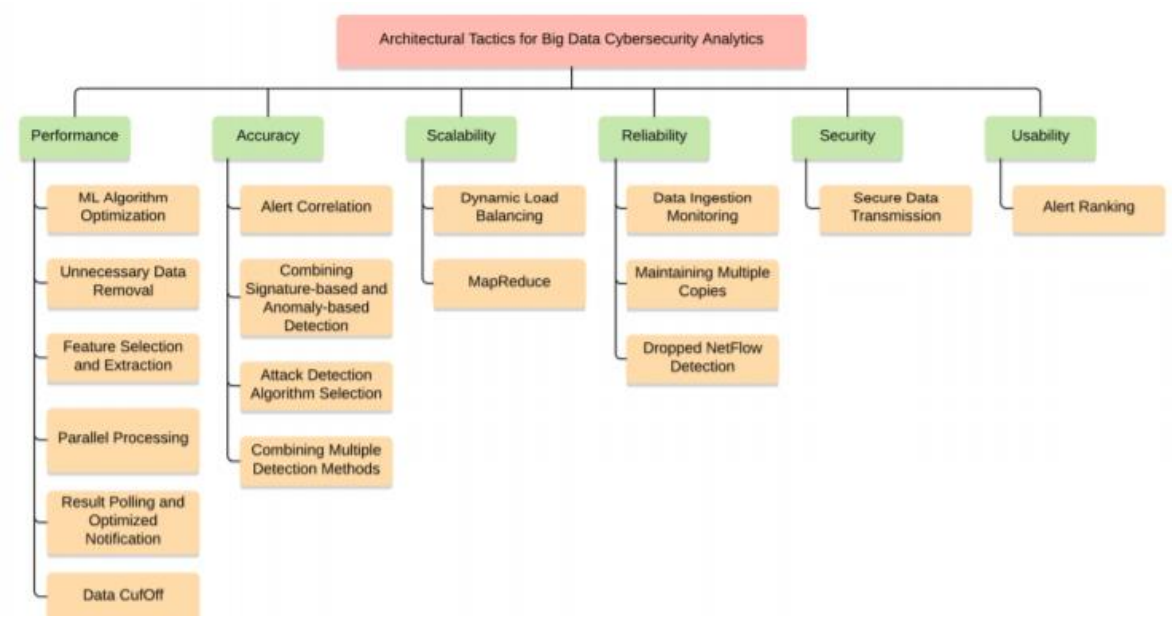


Рисунок 3.1. Архітектурна тактика для систем аналітики безпеки, класифікована на основі відповідних атрибутів якості

3.1.1 Продуктивність

У цьому розділі описується архітектурна тактика, пов'язана з атрибутом продуктивності роботи.

3.1.1.1 Оптимізація алгоритму ML

Вступ: Тактика оптимізації алгоритму ML виявлена у всіх безпеко аналітичних системах, так як всі ці системи безпеки використовують якийсь алгоритм

машинного навчання (ML) або аналізують дані події безпеки. Мета цієї тактики полягає в тому, щоб виділити роль алгоритмів в підвищенні продуктивності системи і дати деякі рекомендації з вибору алгоритму, який найбільш ефективний з точки зору обчислювальної складності.

Мотивація. Двома найбільш важливими факторами, пов'язаними з ефективністю аналітичної системи безпеки, є тип вхідних даних і використовується алгоритм ML. Доступний ряд алгоритмів ML, які можна використовувати в аналітичній системі безпеки. Ці алгоритми ML варіюються від контрольованого навчання (наприклад, логістичної регресії, підтримки векторної машини, Naive Bayes, Random Forest і дерев прийняття рішень) до неконтрольованих алгоритмів навчання (наприклад, K-засобів і нейронних мереж). Ці чинники включають тимчасову складність, інкрементного можливість поновлення, автономний / онлайн-режим і узагальнююча здатність алгоритму, і, найголовніше, вплив алгоритму на швидкість виявлення (точність) системи. Через різноманітної ролі алгоритму, досить складно вибрати найбільш підходящий і ефективний алгоритм.

Опис. Основні компоненти тактики оптимізації алгоритму ML показані на рисунку 3.2. Компонент збору даних збирає дані про події безпеки для навчання аналітичної системи безпеки. Дані навчання можуть збиратися з джерел всередині підприємства, де передбачається розгортання системи, як показано на рисунку 3.2, або вже наявний набір даних, такий як KDDcup99, може використовуватися в якості набору навчальних матеріалів. Після збору даних навчання компонент підготовки даних готує дані для навчання моделі шляхом застосування різних фільтрів. Потім обраний алгоритм ML застосовується до підготовленим даними тренінгу для навчання моделі виявлення атаки. Час, що витрачається алгоритмом на навчання моделі (т. Е час навчання), варіюється від алгоритму до алгоритму. Як тільки модель буде навчена, вона буде перевірена, щоб дослідити, чи може модель виявляти кібератаки. Для тестування моделі дані збираються з підприємства, як показано на кроці 4. Дані тестування фільтруються через компонент підготовки даних і подаються в модель виявлення атаки, яка аналізує дані для виявлення атак на основі

правил, отриманих на етапі навчання. Час, що витрачається моделлю виявлення атаки, щоб вирішити, чи залежить конкретний потік даних від атаки (тобто час прийняття рішення) від використовуваного алгоритму. Результат аналізу даних відображається користувачеві через компонент візуалізації.

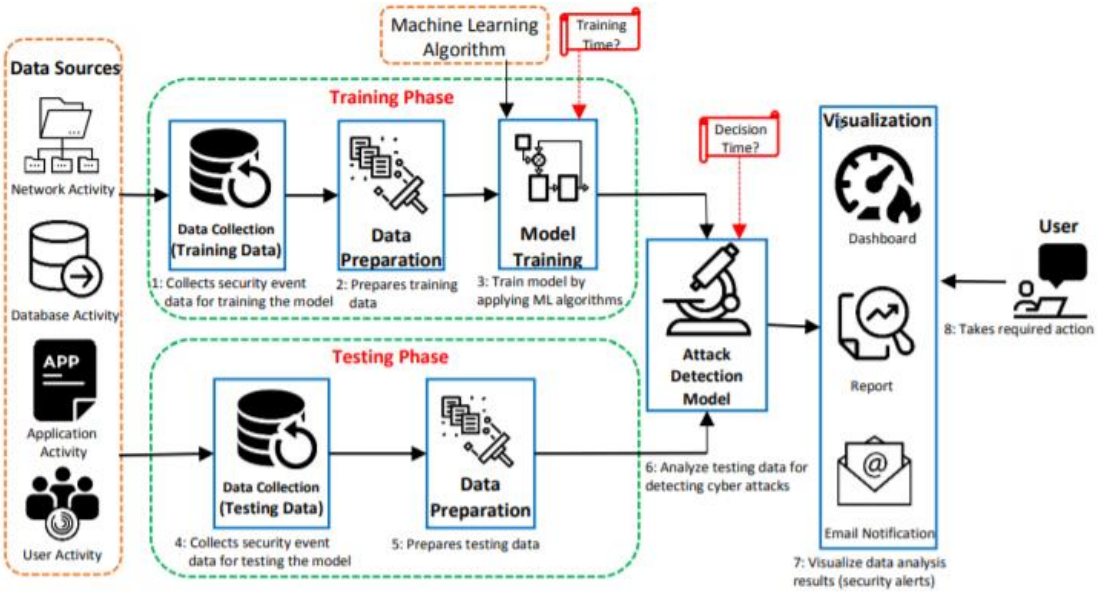


Рисунок 3.2. Стратегія оптимізації алгоритму

Архітектору програмного забезпечення необхідно стежити за декількома речами під час вибору і включення оптимізованого алгоритму.

• Емпіричне порівняння алгоритмів ML передбачає, що продуктивність алгоритмів ML несумісна в різних проблемних областях. Тому алгоритм ML може добре працювати для одного типу аналітики безпеки (наприклад, для виявлення DoS-атаки), але може погано працювати в аналізі безпеки іншого типу (наприклад, виявляти атаку грубої сили).

• Вибір алгоритму є складним в тому сенсі, що на додаток до продуктивності він впливає на якості іншої системи, такі як точність, складність і зрозумілість кінцевого результату. Наприклад, Cheng et al. порівнюють SVM з Extreme Learning Machine (ELM) з точки зору точності і продуктивності. З'ясувалося, що SVM генерує більш точні результати, але є дорогим. З іншого боку, ELM генерує менш точні результати, але має більш легку вагу. При виборі алгоритму повинен бути встановлений розумний компроміс між різними якостями системи.

Вибір алгоритму також залежить від режиму роботи (онлайн або офлайн) аналітичної системи безпеки. Як правило, алгоритми, які мають тимчасову складність, меншу, ніж O (n3), вважаються прийнятними для онлайн-режиму, тоді як алгоритми зі складністю O (n3) і вище є більш повільними і підходять тільки для режиму автономного аналізу. Приклад. Однак, як уже згадувалося, всі наші включені в систему системи використовують алгоритми ML, тут ми повідомляємо результати декількох робіт, щоб продемонструвати роль оптимізованого алгоритму в поліпшенні продуктивності аналітичної системи безпеки.

• Вихідна IDS-платформа: ця аналітична система безпеки порівнює час навчання і час прийняття рішення по п'яти алгоритмам ML, а саме: логістична регресія, векторна машина підтримки, випадковий ліс, дерева прийняття рішень з градієнтом і Naive Bayes. З набором даних KDDCup99 Наївне Байес показує найкращий час тренування (т. Е 79,5 сек), а SVM показує найгірший час тренування (т. Е 479,12 сек). З іншого боку, щодо часу рішення SVM показує найкращий час вирішення (т. Е 10 секунд), а дерево рішень зі збільшеним градієнтом показує найгірший час вирішення (т. Е 22,2 сек).

• Ultra-High-Speed IDS: тут аналітична система безпеки тестується з використанням шести алгоритмів ML для дослідження часу навчання і часу прийняття кожного алгоритму. Алгоритми - це Naive Bayes, SVM, Conjunctive rule, Random Forest, J48 і RepTree. Виявлено, що як з точки зору часу навчання, так і часу прийняття рішення RepTree є найбільш ефективним, за яким слід J48.

• Хмарний детектор загроз: система була реалізована з використанням двох алгоритмів ML - kmean і Naïve Bayes, для вивчення часу навчання, яке взяли обидва алгоритму для навчання системи. Відзначається, що при 500 GB навчальних даних k-засіб займає близько 60 секунд, в той час як Naive Bayes займає близько 92 секунд для навчання моделі.

Залежності: Тактика оптимізації алгоритму ML вимагає застосування тактики тактовності непотрібних даних і тактики вибору і вилучення даних, щоб допомогти зібрати зібрані дані в витончену форму. Після застосування цієї тактики тактика ML

Algorithm Optimization може ефективно застосовуватися до уточненими даними для швидкої підготовки системи і виявлення атак. Тактика оптимізації алгоритму ML повинна бути включена разом з тактикою вибору алгоритму виявлення атаки, так як ці дві тактики встановлюють компроміс між ефектами алгоритму ML по продуктивності і точності.

3.1.1.2. Паралельна обробка

Вступ: Тактику паралельної обробки можна знайти у всіх розглянутих дослідженнях. Ця тактика розподіляє обробку великої кількості даних подій безпеки між різними вузлами обчислювального кластера. Вузли обробляють дані паралельно, що значно покращує час відгуку системи.

Мотивація: Усередині підприємства є ряд джерел, які генерують дані подій безпеки. Ці джерела включають, але не обмежуються ними, мережеві пристрої (наприклад, комутатори і маршрутизатори), операції з базами даних, дані додатків і дії користувача. Дані про події безпеки генеруються з дуже високою швидкістю. Наприклад, підприємство, таке ж велике, як HP, створювало близько трильйона подій безпеки в день в 2013 році, і очікується, що в найближчі роки він буде рости. Окремий комп'ютер, який обробляє такий великий розмір даних подій безпеки послідовним чином, займе багато часу, щоб виявити атаку, яка недопустима в таких критично важливих ситуаціях.

Опис: На рисунку 3.3 показані основні компоненти тактики паралельної обробки. Цифри на малюнку показують послідовність операцій. Компонент збору даних збирає дані подій безпеки з різних джерел в залежності від типу аналітики безпеки і вимог безпеки підприємства. Складальник даних направляє зібрані дані в компонент зберігання даних, який зберігає дані. Дані можуть зберігатися кількома способами, такими як система розподіленої файлової системи Hadoop (HDFS), HBase і реляційна база даних (RDBMS). Для забезпечення паралельної обробки збережені дані необхідно розділити на блоки фіксованого розміру (наприклад, 64

МБ або 128 МБ). Після секціонування дані обробляються в компоненті аналізу даних через кілька складових, які працюють паралельно відповідно до принципів розподіленої структури, такими як Hadoop або Spark. Результат аналізу ділиться з користувачем через компонент візуалізації

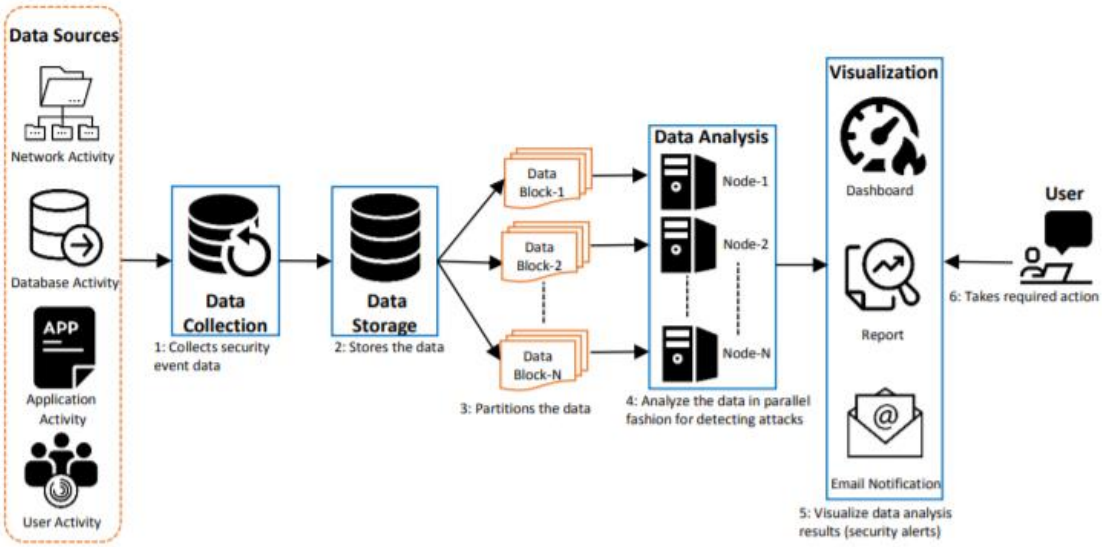


Рисунок 3.3. Тактика паралельної обробки

Тактика паралельної обробки передбачає, що аналітична система безпеки, що включає цю тактику, вже інтегрована з кластером вузлів, здатних обробляти дані паралельно. Іншим важливим фактором, який необхідно подбати, є порушення логічної запису через два блоку при поділі даних на блоки. У такій ситуації важливо зберегти достатню інформацію про тип даних файлу, щоб запис могла бути відновлена.

Приклад. Виявлення фішингу на основі Honeypot демонструє придатність цієї тактики для поліпшення часу відгуку системи виявлення фішингових атак. Автори порівняли час відгуку послідовно реалізованої системи з паралельною системою, кожна з яких обробляє 268 ГБ даних подій безпеки. це було

що для послідовної реалізації системи знадобилося 180 хвилин для обробки всіх даних. З іншого боку, паралельна реалізація системи з фреймами Hadoop і Spark зайняла 21 хвилину і 14 хвилин відповідно з кластером з 9 вузлів. Автори також продемонстрували, що чим більше число вузлів в сценарії паралельної обробки, тим

швидше буде час відгуку системи. Наприклад, час відповіді Hadoop було записано як 57, 36 і 21 хвилин з 3, 5 і 9 вузлами відповідно.

Залежності. Тактика паралельної обробки залежить від тактики балансування динамічного навантаження і тактирования контролю за прогласованностью даних для балансування навантаження між вузлами і управління потоком даних у вузлах відповідно.

Приклад. Виявлення фішингу на основі Honeypot демонструє придатність цієї тактики для поліпшення часу відгуку системи виявлення фішингових атак. Було зіставлено час відгуку послідовно реалізованої системи з паралельною системою, кожна з яких обробляє 268 ГБ даних подій безпеки. Було виявлено, що для послідовної реалізації системи знадобилося 180 хвилин для обробки всіх даних. З іншого боку, паралельна реалізація системи з фреймами Hadoop і Spark зайняла 21 хвилину і 14 хвилин відповідно з кластером з 9 вузлів. Автори також продемонстрували, що чим більше число вузлів в сценарії паралельної обробки, тим швидше буде час відгуку системи. Наприклад, час відповіді Hadoop було записано як 57, 36 і 21 хвилин з 3, 5 і 9 вузлами відповідно.

Залежності. Тактика паралельної обробки залежить від тактики балансування динамічного навантаження і тактирования контролю за прогласованностью даних для балансування навантаження між вузлами і управління потоком даних у вузлах відповідно.

3.1.2. Точність

У цьому розділі описується архітектурна тактика, пов'язана з атрибутом якості точності.

3.1.2.1. Кореляція сповіщень

Вступ: Alert. Ця тактика аналізує окремі попередження, що створюються системами безпеки, відкидає несуттєві попередження і групує разом відповідні

попередження на основі логічних відносин між ними для забезпечення глобального і стисненого уявлення про обстановку безпеки інфраструктури організації. Включення цієї тактики підвищує точність аналітичної системи безпеки, зменшуючи кількість помилкових спрацьовувань і виявляючи складні і складні атаки.

Мотивація: Організації використовують різні засоби і технології безпеки для кращого виявлення своїх мереж і хостів. Наприклад, типова організація може розгортати брандмауер, антивірус і IDS для прийняття / видалення мережевого трафіку, сканувати шкідливе ПО на основі визначеної підписи і виявляти відомі шаблони атаки або ненормальна поведінка відповідно. На жаль, ці засоби безпеки генерують велику кількість попереджень. Наприклад, IDS, розгорнута в реальній мережі, генерує близько 9 мільйонів попереджень в день. Розслідування та реагування на ці численні попередження досить складно, особливо коли 99% з них є помилкові спрацьовування. Крім того, ці кошти забезпечення безпеки ізолюють безпеку без урахування контексту і логічного зв'язку між попередженнями, створюваними іншими інструментами безпеки. Цей ізольований підхід не здатний виявляти атаки, які працюють в повільному режимі протягом певного періоду часу, коли деякі попередження є попередниками більш складних і небезпечних атак. Для вирішення цієї проблеми потрібно керувати на високому рівні, яке пов'язує попередження, беручи до уваги контекст і їх логічні відносини, перш ніж повідомляти про це користувачам.

Опис: На рисунку 3.4 показані основні компоненти, задіяні в тактиці кореляції Alert. Модуль збору даних збирає дані подій безпеки з різних джерел. Зібрані дані зберігаються в сховищі даних і копіюються в компонент попереднього процесора даних для попередньої обробки необроблених даних. Попередньо оброблені дані потрапляють в компонент аналізу попереджень, який аналізує дані для виявлення атак. Варто зазначити, що компонент аналізу оповіщення аналізує дані ізольованим чином (без урахування будь-яких контекстної інформації) або з використанням аналізу, заснованого на використанні, або на основі аномалій, або на обох.

Згенеровані попередження передаються компоненту перевірки повідомлень, який використовує різні методи для визначення того, чи є попередження помилковим. Попередження, ідентифіковані як помилкові спрацьовування, відкидаються на цьому етапі. Чисті і синтезовані попередження передаються компоненту кореляції сповіщень для подальшого аналізу. Сигнали корелюють (тобто логічно пов'язані) з використанням різних методів, таких як кореляція на основі сценаріїв, кореляція на основі правил, статистична кореляція і тимчасова кореляція. Компонент кореляції Alert координує зі зберіганням даних для отримання необхідної контекстної інформації про попередження. Результати кореляції вивільняються через компонент візуалізації. Нарешті, генерується або автоматичний відповідь, або адміністратор безпеки аналізує загрозу і відповідає відповідно.

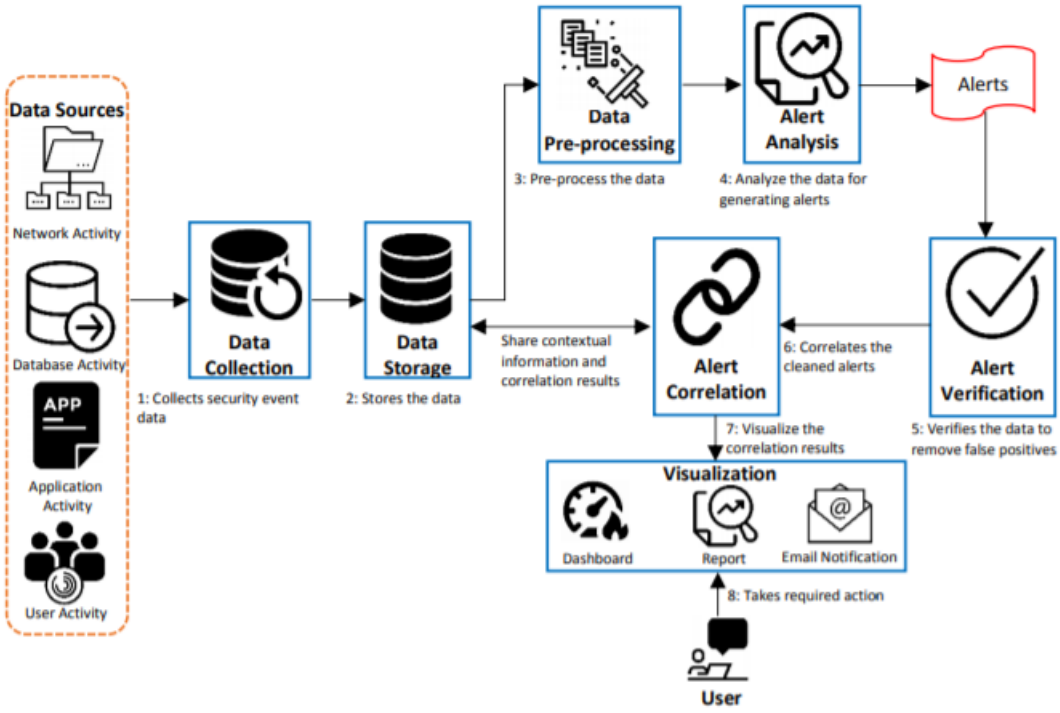


Рисунок 3.4. Тактика сигнальної кореляції.

Поглиблений аналіз, який використовується для корекції опори, підвищує точність, але збільшує час відгуку через включення в систему додаткового складного етапу обчислення. Тактика також вимагає механізмів для придбання знань про доменах і адаптації до змін в мережах і інфраструктурі хоста. Приклади. У дев'яти системах, що реалізують тактику кореляції сповіщень, збираються окремі оповіщення з різних систем безпеки (наприклад, IDS або брандмауер) і корелюють

їх для виявлення атак. Те, що змінюється між цими дев'ятьма системами, є використовуваним методом кореляції. Ми висловлюємо основні методи кореляції для деяких з них.

• GSLAC: система використовує метод причинно-наслідкового зв'язку для кореляції попереджень. Кожне попередження розглядається як вектор з декількома атрибутами (тобто IP-адреса призначення, IP-адреса джерела, тимчасова мітка і т. Д.). Попередження представлені у вигляді графіка, в якому кожне попередження має попереднє попередження і наслідок. Аналітик безпеки аналізує графік для визначення складних сценаріїв атаки.

• Мисливські атаки в темряві: ця система корелює оповіщення на основі їх IP-адрес джерела і одержувача. Подібність між IP-адресами для різних попереджень вимірюється з використанням потужностей перетинів, і якщо оцінка подібності перевищує визначений поріг, це означає, що оповіщення відносяться до одного і того ж IP, які сигналізують про потенційну атаку.

• Багаторівневий коррелятор попереджень: ця система використовує модель передумов і наслідків, пропонованих для визначення взаємозв'язку між окремими попередженнями. Попередження корелюються на основі подібності між вихідним IP-адресою, IP-адресою призначення, часом початку і часом закінчення. створюється граф який показує кожне попередження з його передумовою і наслідком. Якщо попередня умова попередження присутній на графіку в результаті попереднього попередження, ці два попередження тісно пов'язані і аналізуються для опису складного сценарію атаки.

Залежності: Тактика Alert Correlation буде корелювати оповіщення будь-якої якості; проте ефективна кореляція вимагає, щоб попередження були хорошої якості, які залежать від тактики, застосовуваної в модулі аналізу даних, такий як тактика вибору алгоритму атаки і комбінованого виявлення на основі сигнатур і аномалій.

3.1.2.2. Об'єднання виявлення на основі сигнатур і аномалій

Вступ: Методика виявлення комбінованою підписи і аномалії дозволяє аналітичній системі безпеки аналізувати зібрані дані подій безпеки для двох цілей: (1) знайти збіг з уже наявними шаблонами атаки або сигнатурами і (2) знайти відхилення від дізнався нормальна поведінка, тобто аномалію. У будь-якому випадку, знайшовши збіг або відхилення, система генерує сигнальну сигналізацію про можливу кібер-атаці. Комбінація неправильного використання і аномалії значно покращує точність виявлення і зменшує неправдиву позитивну швидкість.

Мотивація: Грунтуючись на принципі виявлення, аналітичні системи безпеки бувають двох типів: засновані на сигнатурі (часто звані несумісними) і засновані на аномалії. Системи на основі сигнатур виявляють атаки на основі визначених шаблонів атаки. Ці шаблони розроблені на основі вже повідомляються атак. Якщо поточна діяльність відповідає шаблоном атаки, ця діяльність називається зловмисної. Ці типи систем дуже ефективні при виявленні відомих атак, але не можуть виявити невідомі атаки. Системи, засновані на аномалії, вивчають нормальна поведінка інфраструктури організації, і будь-яка діяльність, яка відрізняється від цієї поведінки, називається атакою. Цей клас систем може виявляти невідомі атаки, проте він генерує велику кількість помилкових позитивних сигналів. З огляду на обмеження обох типів систем, важливо придумати рішення, яке може мінімізувати ці обмеження.

Опис. Основними компонентами тактики виявлення комбінованою підписи і аномалії є: показаний на рисунку 3.5. Компонент збору даних збирає дані, що відносяться до безпеки, з різних джерел. Зібрані дані зберігаються компонентом зберігання даних. Потім дані вводяться в модуль виявлення на основі сигнатур, який аналізує дані для ідентифікації шаблонів атаки. Для такого аналізу цей компонент використовує заздалегідь розроблені правила з бази даних правил, які визначають шаблони атаки. Якщо збіг ідентифіковано, попередження створюється безпосередньо через компонент візуалізації. Якщо модуль виявлення на основі сигнатури не може виявити ніякого шаблону атаки в даних, дані пересилаються в модуль виявлення аномалій для виявлення невідомих атак, які не можуть бути

виявлені модулем виявлення на основі сигнатур. Компонент виявлення аномалій аналізує дані з використанням алгоритмів машинного навчання для виявлення відхилень від нормальної поведінки. Коли виявлена аномалія (відхилення), через компонент візуалізації створюється попередження. У той же час аномалія визначається у вигляді правила або шаблону атаки і додається в базу даних правил. Таким чином, база даних правил постійно оновлюється, щоб дозволити модулю виявлення підписи виявляти різні атаки.

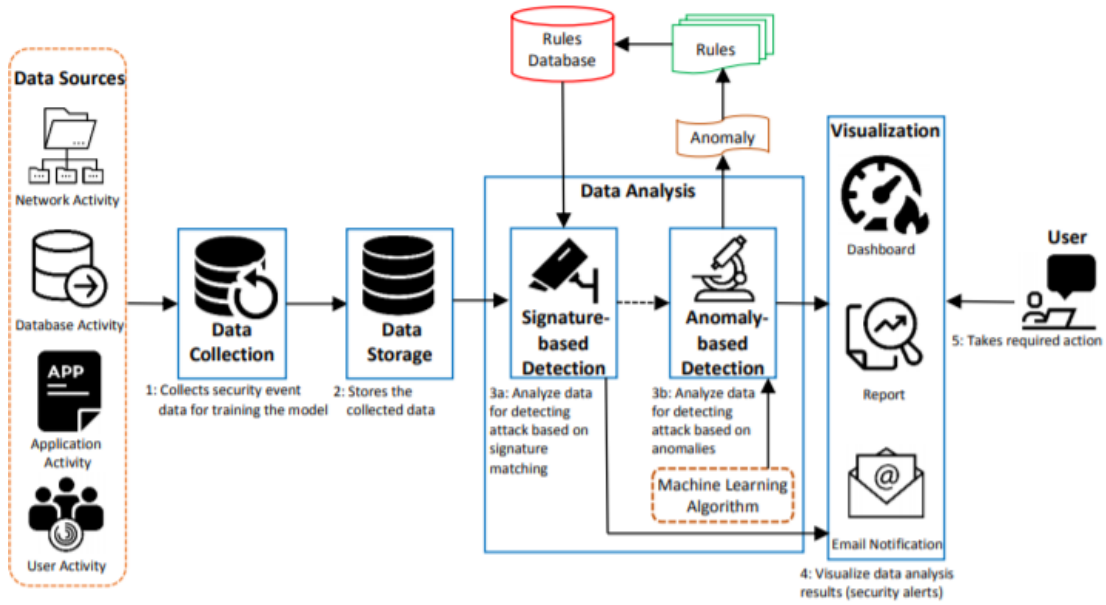


Рисунок 3.5. Об'єднання тактики виявлення на основі сигнатур і аномалій

Об'єднання тактики виявлення на основі сигнатур і аномалій може вплинути на загальну продуктивність системи, ввівши додаткові аналітичні вимоги до даних. Крім того, об'єднання детекторів на основі сигнатур і аномалій в єдину аналітичну систему безпеки і забезпечення їх взаємодії ефективним і успішним способом може бути більш складним і складним.

Приклад: Наступні дві системи демонструють реалізацію тактики виявлення комбінованою підписи і аномалії.

• Гібридне виявлення вторгнень: ця система об'єднує сигнатурний IDS (Snort) з аномальним детектором, реалізованим з використанням Hadoop і Hive. Гібридна система порівнюється з автономною системою підпису з точки зору точності. Обидві системи були оцінені за допомогою декількох атак, таких як ICMP, Smurf,

SYN flood, UPD і атака сканування порту. Виявлено, що гібридна система досягає більш високої швидкості виявлення для всіх атак в порівнянні з автономною системою. Наприклад, при атаці сканування портів автономна система досягає швидкості виявлення близько 40%, в той час як гібридна система показує швидкість виявлення близько 50%.

• Snort + PHAD + NETAD: ця система об'єднує IDS (Snort) на основі сигнатур з двома аномальними детекторами, тобто детектором аномалій заголовка пакета (PHAD) і детектором аномалій мережевого трафіку (NETAD). Гібридну систему оцінюють з використанням набору даних IDEVAL для вивчення його швидкості виявлення в порівнянні з автономною системою на основі сигнатур. Спостерігається, що автономна система може виявляти тільки 27 атак, в той час як гібридна система виявляє 146 атак з набору даних.

Залежності: Ця тактика комбінованої підписи і виявлення аномалій вимагає тактики паралельної обробки, щоб зменшити додатковий час аналізу, необхідний через двухфазного аналізу. Крім того, ця тактика також залежить від тактики вибору алгоритму виявлення атаки, щоб ефективно вибирати ефективний алгоритм виявлення аномалій в даних подій безпеки.

3.1.3. Масштабованість

У цьому розділі описується архітектурна тактика, пов'язана з атрибутом якості масштабування.

3.1.3.1. Динамічне балансування навантаження

Вступ: Тактику динамічного балансування навантаження можна знайти в GSLAC і Cloud Bursting. Ця тактика використовується для балансування навантаження на обробку серед вузлів аналізу шляхом ділення даних подій безпеки між вузлами. Маючи пропускну здатність, доступну в кластері, тактика динамічного балансування навантаження робить системний масштаб добре, не додаючи додаткових апаратних ресурсів.

Мотивація: Аналітична система безпеки використовує кластер обчислювальних вузлів для зберігання і аналізу даних подій безпеки. Розмір кластера відрізняється в різних системах (наприклад, 10 вузлів або 5). Система розподіляє дані події безпеки між вузлами для прискорення процесу. Коли швидкість введення даних зростає (наприклад, з 100 МБ / с в будні дні до 150 МБ / с у вихідні дні), важливо, щоб система розподіляла збільшене навантаження збалансованим чином, щоб уникнути ситуації, коли один вузол знаходиться в екстремальному стані (т. е. 100% завантаження ЦП), а інший вузол завантажений (т. е. 30% завантаження ЦП).

Опис: На рисунку 3.6 показані основні компоненти тактики динамічного балансування навантаження. Компонент збору даних збирає дані з різних джерел. Захоплені дані відправляються компоненту фільтрації даних, який видаляє дані, які не сприяють процесу виявлення атак. Відфільтровані дані пересилаються в балансувальник навантаження, який розподіляє дані між різними вузлами для балансування робочого навантаження між вузлами модуля аналізу даних. Дані можуть бути розподілені на основі різних критеріїв. Наприклад, мережевий трафік може бути розподілений на основі інформації заголовка (наприклад, IP-адреси або TCP-портів) або інформації корисного навантаження (наприклад, підписи). Збалансований розподіл даних не є повністю надійним через різнорідності даних подій безпеки. Наприклад, якщо мережевий трафік розподіляється на основі діапазону IP, один діапазон може містити більше кількості пакетів з великим розміром, які можуть вичерпувати один вузол. З іншого боку, вузол, керуючий іншим діапазоном IP з невеликою кількістю пакетів невеликого розміру, може сидіти без діла. Тому вводиться компонент моніторингу ресурсів, який постійно відстежує використання ЦП вузлів і звітів в балансувальник навантаження. Коли різниця між завантаженням процесора вузлами перетинає зумовлений поріг, балансувальник навантаження балансує навантаження між вузлами, переміщаючи навантаження обробки з перевантажених вузлів на менш завантажені вузли. Після успішного аналізу даних в модулі аналізу даних результати будуть спільно використовуватися користувачем через компонент візуалізації.

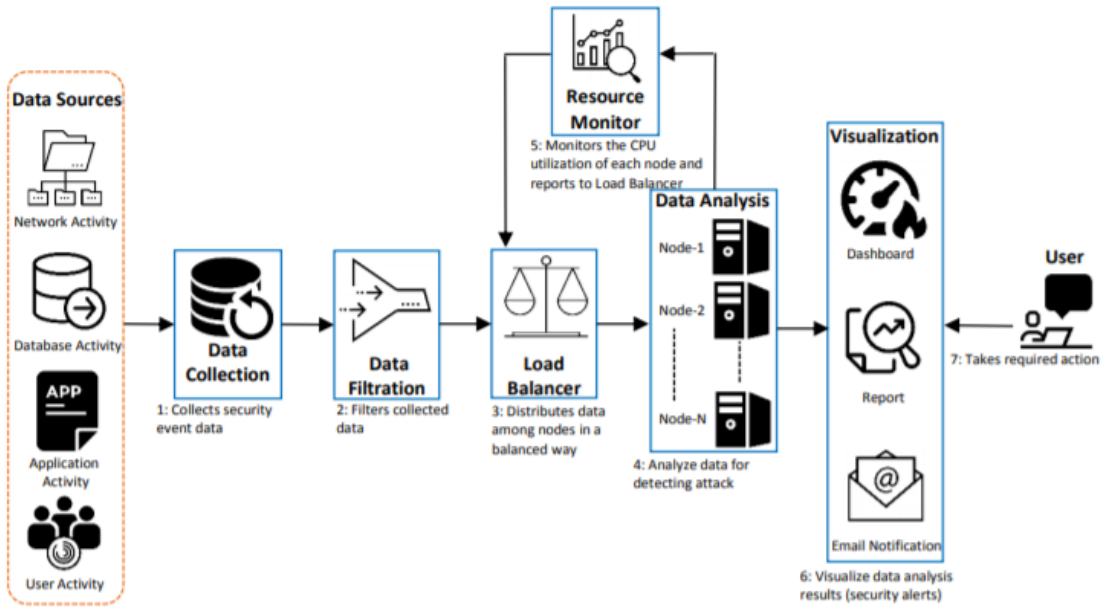


Рисунок 3.6. Динамічна тактика балансування навантаження

Ця тактика передбачає, що кластер має обчислювальну потужність; В іншому випадку немає ніякої можливості балансувати навантаження, якщо всі вузли вже використовують 100% потужності процесора. Крім того, ця тактика вимагає ефективного механізму для вибору цільового вузла і визначення кількості даних, які повинні бути переміщені з перевантаженого вузла в цільової (недовантажений) вузол.

Приклади. У наступних системах реалізована тактика динамічного балансування навантаження.

• GSLAC: під час роботи монітор ресурсів контролює робоче навантаження вузлів і періодично оновлює список латентності, що показує використання ЦП вузлів. Коли різниця між завантаженням процесора між вузлами перетинає порогове значення, монітор ресурсів сигналізує про балансуванню навантаження. Балансувальник навантаження використовує динамічні міграції гарячих точок для балансування робочого навантаження між вузлами.

• Cloud Bursting: ця система використовує тактику динамічного балансування навантаження трохи інакше. Монітор ресурсів контролює робоче навантаження локального кластера вузлів і продовжує надавати інформацію про стан балансувальник навантаження. Після прибуття нового потоку даних подій безпеки

балансувальник навантаження розглядає доцільність внесення запускати роботу з аналізу даних локально або розбити його на інші кластери в хмарі.

Залежності. В принципі, тактика динамічного балансування навантаження не вимагає ніякої іншої тактики для її реалізації. Однак в аналітичній системі безпеки він буде працювати в координації з тактикою, як «Видалення непотрібних даних», «Вибір і витяг функцій», тактика Data CutOff і паралельна обробка.

3.1.3.2. MapReduce

Вступ: Ця тактика була виявлена у всіх розглянутих системах, які використовують платформу Hadoop для зберігання і аналізу даних подій безпеки. Тактика MapReduce забезпечує структуру програмування для масштабування програмних додатків в кластері з декількох вузлів. Хоча MapReduce - добре документована структура програмування, ця тактика відображає внесок MapReduce, що відноситься до підвищення масштабованості. Тактика вирішує різні проблеми масштабованості, які включають в себе складні механізми розпаралелювання, синхронізації і зв'язку. Крім безпеки, MapReduce також є широко поширеною структурою в інших областях великий аналітики даних, таких як біоінформатика, астрономія і охорону здоров'я.

Мотивація: Щоб масштабувати аналітичну систему безпеки для обробки величезного обсягу даних, простим рішенням є додавання додаткового обладнання в систему. Додаткове обладнання може бути додано на одну і ту ж фізичну машину (вертикальне масштабування) або може бути додано як окрема фізична машина (горизонтальне масштабування). Дуже складно ефективно обробляти складну розпаралелювання, синхронізацію, зв'язок і використання ресурсів з додаванням апаратних ресурсів в існуючу систему. Тому для вирішення цих невирішених проблем необхідна структура.

Опис: Основні компоненти тактики MapReduce показані на рисунку 3.7. Компонент збору даних збирає дані події безпеки з одного або декількох джерел.

Зібрані дані передаються компоненту фільтрації даних для видалення даних, які не сприяють виявленню атак. Відфільтровані дані розбиваються на головні вузли для зберігання в файлах HDFS вузлів даних. Перетворювач всередині кожного вузла даних зчитує свій призначений блок HDFS даних для обробки. Варто зазначити, що кількість перетворювачів не залежить від кількості вузлів даних, а залежить від кількості блоків вхідних даних. Маппера обробляють дані одночасно в формі пар ключ-значення (ключ, значення) для генерації проміжних результатів (ключ, список значень). Проміжні результати сортуються, упорядковано і передаються в редуктори, які об'єднують і агрегує проміжні результати для отримання кінцевого результату. Кількість одночасно працюючих сканерів і редукторів залежить від потужності, робочого навантаження і рекомендацій користувачів. Якщо система відчуває підвищений введення даних, додаткові апаратні ресурси (вузли даних) можуть бути легко додані в кластер для обробки збільшеною робочого навантаження. Для більш докладної інформації про архітектуру Hadoop і MapReduce читачі можуть проконсультуватися з цими порадами.

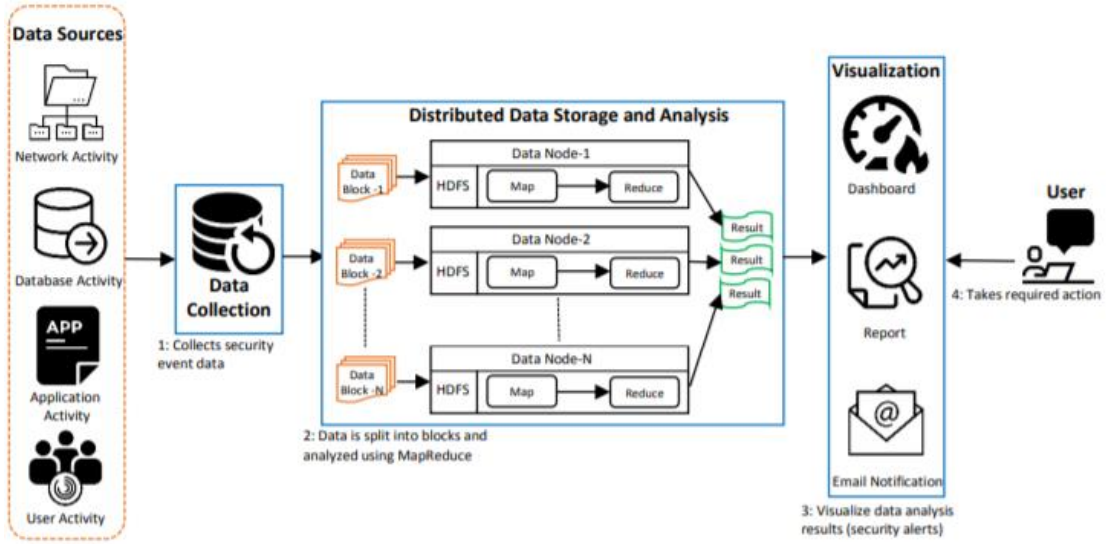


Рисунок 3.7. Тактика MapReduce

Ця тактика передбачає, що для горизонтального масштабування системи є додаткове обладнання. Крім того, введення накладних витрат, викликаних операціями читання / запису на диску, є актуальною проблемою з MapReduce, що може значно продовжити час відгуку аналітичної системи безпеки. Приклад.

Тактика MapReduce використовується поряд розглянутих систем, однак ми опишемо деякі з них, щоб проілюструвати її внесок в досягнення масштабованості.

• Вимірювання трафіку і аналіз за допомогою Hadoop: ця система використовує евристичний алгоритм, який дозволяє картографам зчитувати записи пакетів з HDFS на основі біта мітки часу заголовка пакета. Щоб оцінити масштабованість, 1 ТБ даних подій безпеки аналізується кластером з різною кількістю вузлів від 5 до 30. Спостерігається, що продуктивність системи збільшується пропорційно розподілу ресурсів. Наприклад, час завершення аналізу зменшується з 71 хвилини з п'ятьма вузлами до майже 36 хвилин з 10 вузлами.

• IDS-MRCPSO: ця система використовує алгоритм кластеризації оптимізації ройових частинок за допомогою MapReduce. Щоб дослідити масштабованість, система реалізується з різною кількістю вузлів. Виявлено, що час прискорення лінійно зростає на початку від 2 до 8 вузлів, але починає відхилятися від лінійного при переміщенні від 8 до 16 вузлів. Це відхилення пов'язане з каркасом Hadoop, тобто з запуском MapReduce і збереженням проміжних результатів.

• Extreme Learning Machine: ця аналітична система безпеки об'єднує можливості лінійного масштабування алгоритму Extreme Learning Machine і MapReduce. Система була реалізована з 15, 20, 25 і 30 вузлами для оцінки її масштабованості. Система показує лінійне масштабування від 15 до 25 вузлів, але відхиляється від лінійного масштабування після 25 вузлів. Автори стверджують, що такий витік виникає через підвищену зв'язку між вузлами.

Залежності. Як така ця тактика не вимагає ніякої іншої тактики для її реалізації.

3.1.4. Надійність

У цьому розділі описується архітектурна тактика, пов'язана з атрибутом якості «Надійність».

3.1.4.1. Тактика моніторингу прийому даних

Вступ: Тактика моніторингу проникнення даних використовується в системах виявлення загроз на основі потоків, виявлення шкідливих IP-адрес і IDS на основі GPGPU. Ця тактика контролює потік даних від збирача даних на обчислювальні сервери в системах потокового відтворення в режимі реального часу. Якщо швидкість притоку даних стає занадто високою, що може привести до збою обчислювального сервера, ця тактика автоматично блокує приплив даних в обчислювальний сервер.

Мотивація: Великий обсяг даних про події безпеки збирається з різних джерел на підприємстві. Ці джерела включають, але не обмежуються, дані мережевого журналу, дані про активність користувача, дані про активність додатки і журнали доступу до хосту. Ці дані великого розміру генеруються з високою швидкістю і передаються безпосередньо в аналітичну систему безпеки, яка полегшує аналітику в реальному часі. Однак іноді дані можуть генеруватися і збиратися зі швидкістю, що перевищує можливості обчислювального кластера для обробки даних, що може привести до збою обчислювального сервера. Отже, необхідна тактика для контролю швидкості генерації даних і контролю потоку даних на обчислювальні сервери. Опис. Основні компоненти тактики Data Ingestion Monitoring показані на рисунку 3.8. Компонент збору даних збирає дані з різних джерел, використовуючи різні інструменти, такі як Wireshark, для збору даних мережевого трафіку. Дані, зібрані різними вузлами модуля збору даних, переміщаються в компонент розподіленого зберігання і аналізу даних через монітор прийому даних. Роль монітора прийому даних полягає в тому, щоб підтримувати потокову передачу даних в реальному часі в розподілений кластер зберігання і аналізу даних. Монітор прийому даних контролює швидкість притоку в розподілений кластер. Якщо швидкість надходження даних з збирача даних стає настільки високою, що вона перевищує можливості обчислювального сервера і може привести до збою обчислювального кластера, монітор заблокує потік даних в обчислювальний кластер. Монітор прийому даних регулює потік даних між компонентом збору даних і компонентом

розподіленого зберігання і аналізу даних. Потік даних про події безпеки, що надходять в компонент розподіленого зберігання і аналізу, аналізується для виявлення кібератак, а результати відображаються користувачеві через компонент візуалізації.

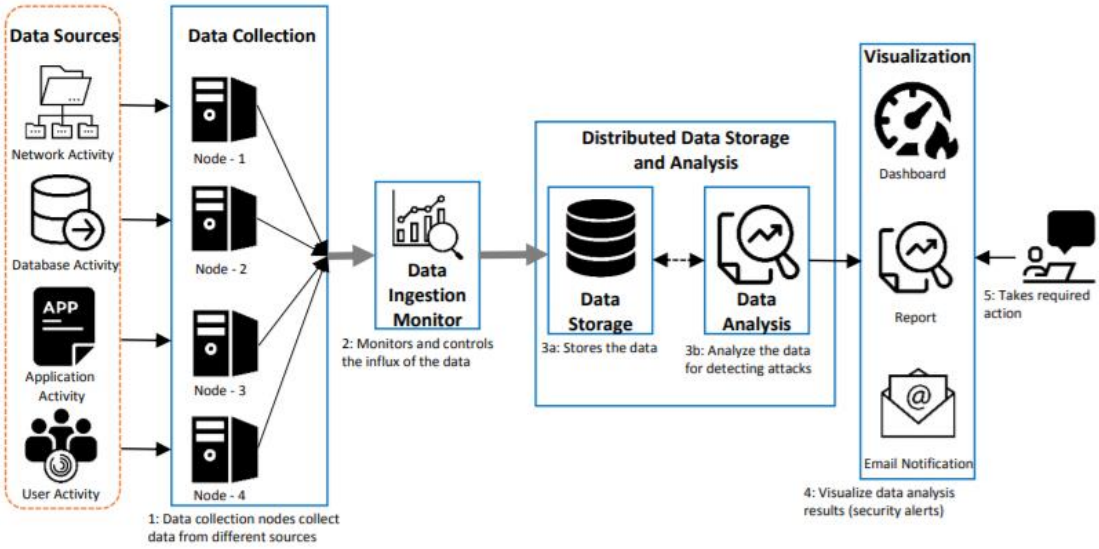


Рисунок 3.8. Тактика моніторингу прийому даних

Обмеження. Тактика моніторингу проникнення даних вимагає досвіду для установки монітора прийому даних, що є складним завданням, особливо в сильно розподіленої установці. Крім того, ця тактика також вимагає великих інвестицій для настройки сервера моніторингу. Ця тактика найкраще підходить для аналітичних систем безпеки, розгорнутих на підприємствах, які з високою швидкістю генерують великий обсяг даних про події безпеки і направляють їх безпосередньо в аналітичний модуль для обробки в реальному часі.

Приклад: Тактика моніторингу проникнення даних включена в наступні системи.

• Виявлення шкідливого IP-адреси: ця система демонструє, як тактика моніторингу проникнення даних запобігає збій обчислювального сервера. У цій системі сервер Flume використовується в якості монітора прийому даних для моніторингу і контролю припливу даних в обчислювальний кластер. Автори оцінили систему з чотирма швидкостями надходження даних (тобто 0,4 мільйона

записів / хв, 0,7 мільйона записів / хв, 0,8 мільйона записів / хв, 0,85 мільйона записів / хв). Повідомлялося, що обчислювальний сервер має пропускну здатність до 0,8 млн. Записів в хвилину, однак при збільшенні швидкості до 0,85 млн. Записів в хвилину комп'ютерний сервер досягає свого максимального межі. Збільшення швидкості припливу даних понад 0,85 мільйонів записів / хв призведе до збою обчислювального сервера. Тому при швидкості 0,85 млн. Записів в хвилину монітор прийому даних контролює подальше збільшення швидкості.

• Система виявлення загроз на основі потоків і IDS на основі GPGPU. Обидві ці системи реалізують тактику моніторингу проникнення даних через сервіс Flume, проте в цих дослідженнях не оцінювався вплив тактики на надійність системи.

Залежності. Тактика моніторингу проникнення даних не залежить від будь-якої іншої тактики; тим не менше, її можна краще консолідувати, якщо вона реалізована разом з тактикою безпечної передачі даних, яка допоможе отримати дані для моніторингу в їх первинному вигляді.

3.3 Архітектура програмної дослідницької системи

Програмна дослідницька система на базі методології паралельних обчислень використовуючи інструменти Hadoop складається з мережі розподілених комп’ютерів (у кількості: 2 ПК та 1 ноутбук), системи виявлення вторгнення в кібербезпеку робототехнічних і автономних систем, платформи Hadoop.

Виконання розподілених задач на платформі Hadoop відбувається в рамках парадигми map/reduce*.

map/reduce – це парадигма (програмна модель) виконання розподілених обчислень для великих обсягах даних.

У загальному випадку, для map/reduce виділяють 2 фази:

▪ map(f, c) *f*, що приймає функцію та список *c*. Повертає вихідний список, який є результатом застосування функції *f* до кожного елемента вхідного списку *c*.

▪ reduce(f, c) приймає функцію *f* та список *c*. Повертає об'єкт, утворений через згортку колекції *c* через функцію *f*.

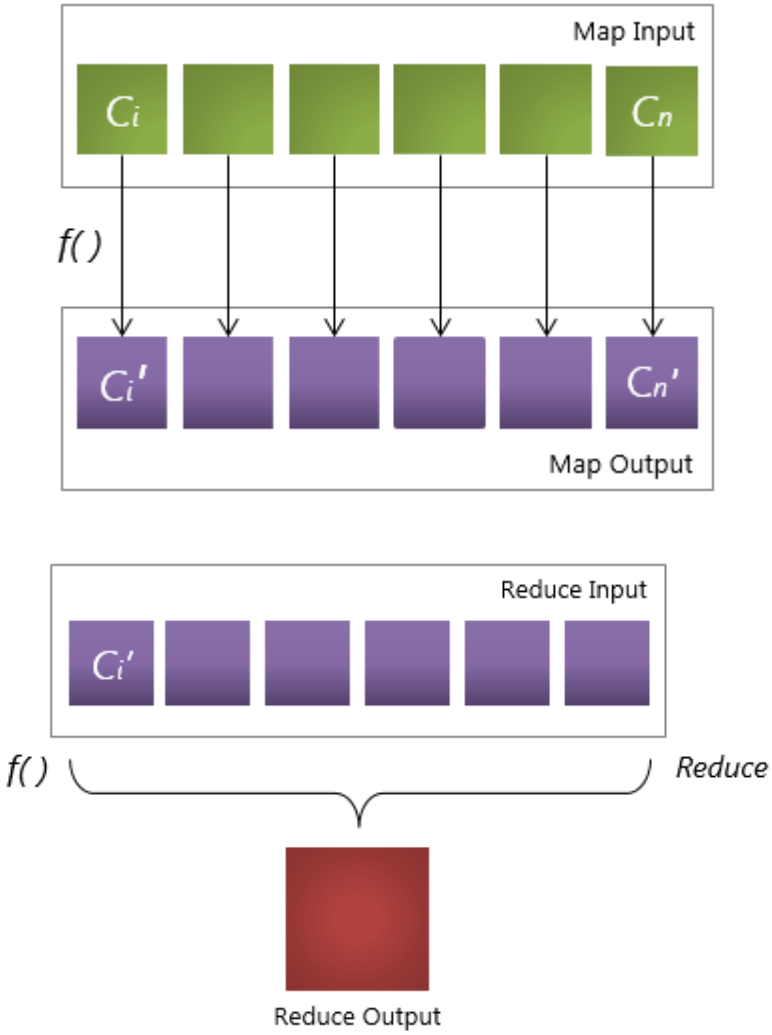


Рисунок 3.9 – Виконання розподілених задач на платформі Hadoop

Програмна модель map/reduce була запозичена з функціонального програмування, хоча в реалізації Hadoop і має певні семантичні відмінності від прототипу в функціональних мовах.

Як і в функціональних мовах, при використанні програмної моделі map/reduce:

▪ вхідні дані не змінюються;

▪ розробник кодує, що потрібно зробити, а не як потрібно зробити.

На січень 2012 року широко відомі наступні програмні реалізації моделі map/reduce:

- *Google MapReduce* – закрита реалізація від Google на C++;
- *CouchDB і MongoDB* – реалізації для NoSQL баз даних;
- *Hadoop MapReduce* – відкрита реалізація на Java для Apache Hadoop.

Hadoop MapReduce – програмна модель (framework) виконання розподілених обчислень для великих обсягів даних в рамках парадигми map/reduce, що представляє собою набір Java-класів і виконуваних утиліт для створення і обробки завдань на паралельну обробку.

Основні концепції Hadoop MapReduce можна сформулювати як:

- обробка/обчислення великих обсягів даних;
- масштабованість;
- автоматичне розпаралелювання завдань;
- робота на ненадійному обладнанні;
- автоматична обробка відмов виконання завдань.

Роботу Hadoop MapReduce можна умовно поділити на наступні етапи:

1. Read Input

Вхідні дані діляться на блоки даних зумовленого розміру (від 16 до 128 Мб Мб) – *сплітu* (від англ. split). MapReduce Framework закріплює за кожною функцією Map певний спліт.

2. Map

Кожна функція Map отримує на вхід список пар «ключ/значення» <k,v>, обробляє їх і на виході отримує нуль або більше пар <k',v'>, що є проміжним результатом.

$$map(k, v) -> [(k', v')]$$

де k' - у загальному випадку довільний ключ, який не збігається з k.

Всі операції map() виконуються паралельно і не залежать від результатів роботи один одного. Кожна функція map() отримує на вхід свій унікальний набір даних, який не повторюється ні для якої іншої функції map().

3. Partition / Combine

Метою етапу *partition* (поділ) є розподіл проміжних результатів, отриманих на етапі map, reduce-завданням.

$$(k', reducers\_count) -> reducer\_id$$

де reducers_count - кількість вузлів, на яких запускається операція згортки;

reducer_id - ідентифікатор цільового вузла.

В найпростішому випадку,

$$reducer\_id = hash(k') \bmod reducers\_count$$

Основна мета етапу partition – це балансування навантаження. Некоректно реалізована функція partition може призвести до нерівномірного розподілу даних між reduce-вузлами.

Функція *combine* запускається після map-фази. В ній відбувається проміжна згортка, локальних по відношенню до функції map, значень.

$$[(k', v')] -> (k', [v'])$$

Основне значення функції combine – комбінування проміжних даних, що в свою чергу веде до зменшення обсягу інформації, що передається між вузлами інформації.

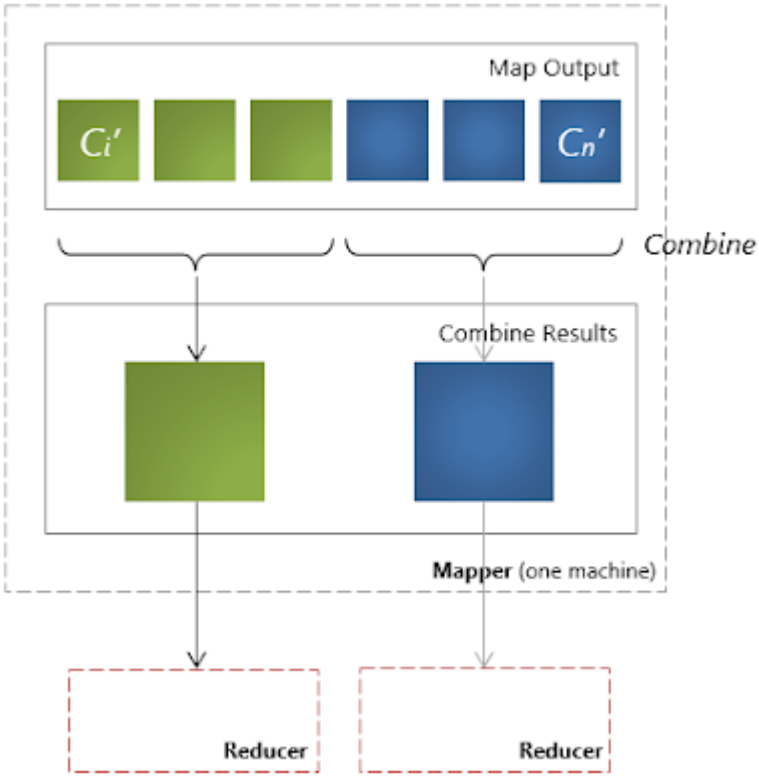


Рисунок 3.10 – Комбінування даних

4. Copy / Compare / Merge

На цьому етапі відбувається:

- *Copy*: копіювання результатів, отриманих в результаті роботи функцій map і combine (якщо така була визначена), з map-вузлів на reduce-вузли.

- *Compare (або Sort)*: сортування, групування за ключем k отриманих в результаті операції copy проміжних значень на reduce-сайті.

$$compare(k'n, k'n + 1) -> \{-1, 0, +1\}$$

- *Merge*: «об'єднати даних, отриманих від різних вузлів, для операції згортки.

5. Reduce

Framework викликає функцію reduce для кожного унікального ключа k' у відсортованому списку значень.

$$reduce(k', [v']) -> [v'']$$

Всі операції reduce() виконуються паралельно і не залежать від результатів роботи один одного. Таким чином, результати роботи кожної функції reduce() пишуться в окремий вихідний потік.

6. Output write

Результати, отримані на етапі reduce, записуються в вихідний потік (у загальному випадку, файлові блоки в HDFS). Кожен reduce-вузол пише у власний вихідний потік.

Всі інші фази виконуються програмною моделлю MapReduce без додаткового кодування з боку розробника. Крім того, середовище виконання Hadoop MapReduce виконує наступні функції:

- планування завдань;
- розпаралелювання завдань;
- перенесення завдань до даних;
- синхронізація виконання завдань;
- перехоплення «провалених» завдань;
- обробка відмов виконання завдань і перезапуск провалених завдань;
- оптимізація мережевих взаємодій.

Архітектура Hadoop MapReduce

Hadoop MapReduce використовує архітектуру «*master-worker*», де master – *єдиний* примірник керуючого процесу (*JobTracker*), як правило, запущений на окремій машині (обчислювальному вузлі). Worker-процеси – це *довільне безліч* процесів *TaskTracker*, які працюють на *DataNode*.

JobTracker і TaskTracker «лежать» над рівнем зберігання HDFS, і запускаються/виконуються у відповідності з наступними правилами:

- примірник JobTracker виконується на NameNode-сайті HDFS;
- примірники TaskTracker виконуються на DataNode-сайті;
- TaskTracker виконуються у відповідності з принципом «*дані близько*», тобто процесTaskTracker розташовується топологічно максимально близько з вузлом DataNode, даніякого обробляються.

Вищеописані принципи розташування JobTracker - і TaskTracker-процесів дозволяють істотно скоротити обсяги переданих по мережі даних і мережеві затримки, пов'язані з передачею цих даних – основні «вузькі місця» продуктивності в сучасних розподілених системах.

JobTracker є єдиним вузлом, на якому виконується програма MapReduce, викликається програмним клієнтом. JobTracker виконує наступні функції:

- планування індивідуальних (по відношенню до DataNode) завдань map і reduce, проміжних згорток;
- координація завдань;
- моніторинг виконання завдань;
- перепризначення завершилися невдачею завдань іншим вузлам TaskTracker.

У свою чергу, TaskTracker виконує наступні функції:

- виконання map - і reduce-завдань;
- управління виконанням завдань;
- відправлення повідомлень про статус завдання та завершення роботи сайту JobTracker;
- відправка діагностичних heartbeat-повідомлень сайту JobTracker.

Взаємодія TaskTracker-вузлів з вузлом JobTracker йде допомогою викликів RPC, причому виклики йдуть тільки від TaskTracker. Аналогічний принцип взаємодії реалізований у HDFS – між вузлами DataNode і NameNode-вузлом. Таке

рішення зменшує залежність керуючого процесу JobTracker від процесів TaskTracker.

Взаємодія JobTracker-вузла з клієнтом (програмним) проходить за наступною схемою: JobTracker приймає завдання (Job) від клієнта і розбиває завдання на безліч M map-завдань і безліч R reduce-завдань. Вузол JobTracker використовує інформацію щодо файлових блоків (кількість блоків і їх місцезнаходження), розташовану у вузлі NameNode, знаходиться локально, щоб вирішити, скільки підлеглих завдань необхідно створити на сайтах типу TaskTracker. TaskTracker отримує від JobTracker список завдань (тасков), завантажує код і виконує його. Періодично TaskTracker відсилає JobTracker статус виконання завдання.

Взаємодії TaskTracker-сайтів із програмним клієнтом відсутні.

За аналогією з архітектурою HDFS, де NameNode є точкою одиничної відмови (*Single point of failure*), JobTracker також є такою.

При збої TaskTracker-сайту JobTracker-вузол перепризначає завдання несправного вузла іншому вузлу TaskTracker. У разі несправності JobTracker-сайту, для продовження виконання MapReduce-додатки, необхідний перезапуск JobTracker-вузла. При перезапуску вузол JobTracker читає зі спеціального журналу дані, щодо останньої успішної контрольної точці (checkpoint), відновлює свій стан на момент запису checkpoint і продовжує роботу з місця останньої контрольної точки.

У розподіленій конфігурації Hadoop є головним (master) вузол і кілька підлеглих (slave) вузлів (рис. 3.11).

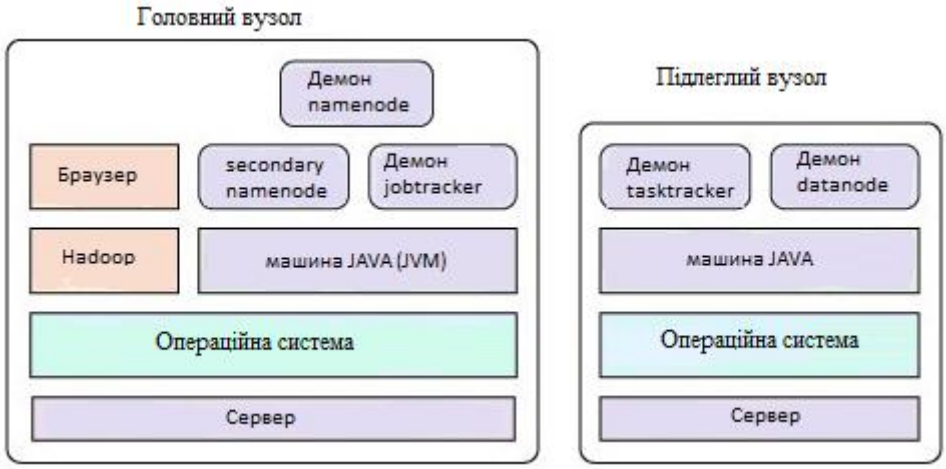


Рисунок 3.11 – Поділ Hadoop на головний і підлеглі вузли

Як показано на рисунку 3.11, на головному вузлі працюють демони namenode, secondarynamenode і jobtracker (так звані *master-демони*). Крім того, з цього вузла здійснюється управління кластером (за допомогою утиліти Hadoop і Web-браузера). На підлеглих вузлах працюють демони tasktracker і datanode (slave-демони). Відмінність даної конфігурації полягає в тому, що на головному вузлі працюють демони, що відповідають за управління та координування кластера Hadoop, тоді як на підлеглих вузлах працюють демони, що забезпечують функції зберігання даних у файловій системі HDFS і реалізують функціонал MapReduce (функція обробки даних).

Для демонстрації цього створимо один головний і два підлеглих вузла, розташовані в одному сегменті локальної мережі. Ця конфігурація зображена на рисунку 3.12.

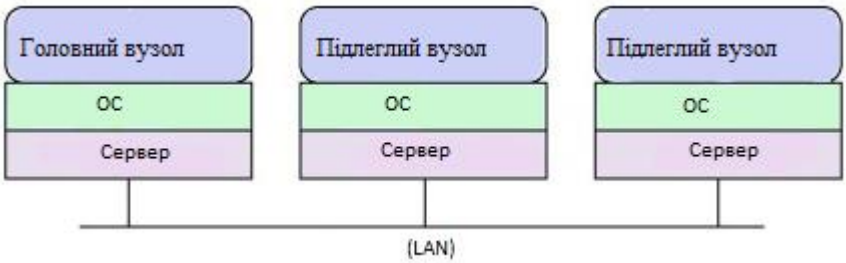


Рисунок 3.12 – Конфігурація кластера Hadoop

Проведемо дослідження програмної дослідницької системи на базі методології паралельних обчислень використовуючи інструменти Hadoop. Також здійснимо порівняння ефективності використання платформи Hadoop та її аналогів.

3.4 Проведення дослідження на базі Hadoop

Для проведення експериментів використовувалася платформа Syncfusion v3.2.0 з двома процесорами Intel CORE I5 @ 2.1 ГГц та операційною системою Windows 10. Машині було виділено 4 thread, зарезервовано (reserved) 2 ГГц процесора, також було зарезервовано 16 ГБ ОЗУ.

В роботі для проведення експериментальної оцінки в цілях підвищення достовірності отриманих результатів використовується вхідний потік даних, зібраний з комп'ютера, що зберігає дані аналізу кібервторгнень на робо технічні та автономні системи. В якості вхідних даних був взятий файл розміром 7 ГБ, в якому містилося приблизно 7 мільйонів подій, що відносяться до 81 типу. Вихідні дані були попередньо зібрані з журналу подій безпеки ОС Windows 10 на офісному комп'ютері за період, приблизно рівний 2 діб.

В ході аналізу вихідних даних було встановлено, що всі типи подій можна розділити на дві групи:

1) часто зустрічаються

2) рідко зустрічаються

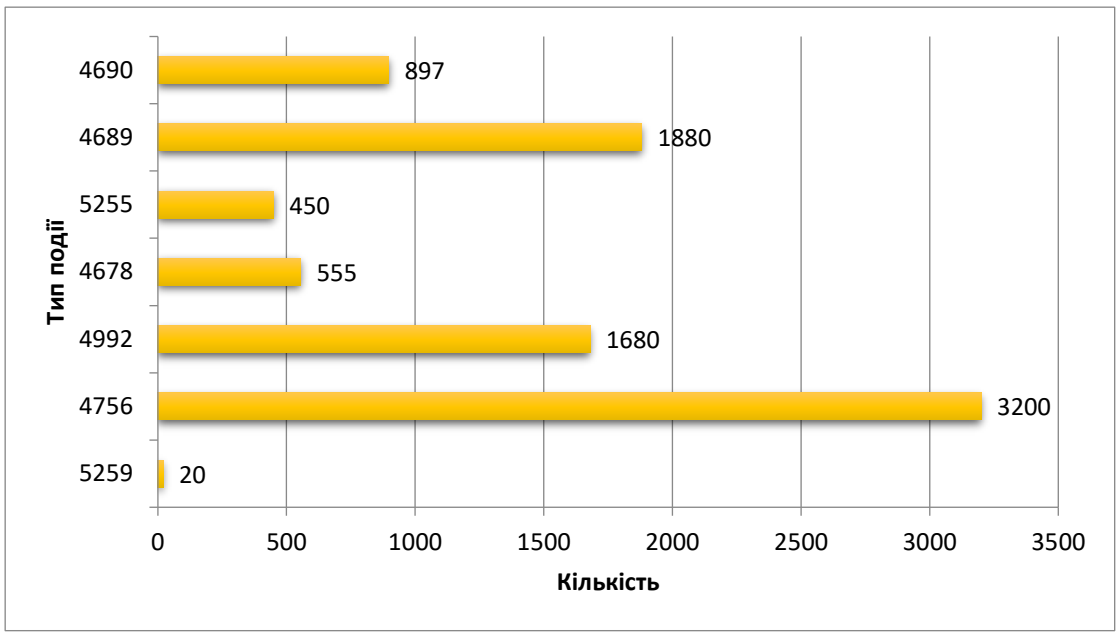


Рисунок 3.13 – Статистика типів подій, які зустрічаються часто

До першої групи належать типи подій зі значенням Count менше 100, до другої-інші типи.

Показники кількості подій першої і другої групи представлені на рис. 3.13 і рис. 3.14 відповідно.

З рис. 3.13 і рис. 3.14 видно, що найбільш складними є обчислення відносних ваг прямих зв'язків між такими типами, як 4689 і 4756. Це призводить до необхідності обробляти 2959 * 1664 пар подій в режимі, близькому до реального, що є надзвичайно складним завданням, якщо не використовувати паралельні обчислення та не обмежувати вибірку, що аналізується.

При виконанні завдання виявлення непрямих зв'язків в ході аналізу вихідних даних було визначено 134 унікальних властивості типів подій. Перетин множин значень даних властивостей дозволило виявити 15 властивостей, що відносяться один до одного як нерівнозначні однотипні. Уточнення графа зв'язків типів подій за рахунок доповнення вузлів відповідними непрямими зв'язками збільшило кількість ребер (загальна кількість зв'язків) приблизно на 20%.

Статистичні підходи при виконанні процесу кореляції подій безпеки застосовувалися раніше. Складність застосування імовірнісної оцінки полягає в

необхідності забезпечення відповідності вихідних даних певним вимогам, наприклад, їх однорідності, нормальному закону розподілу та іншим. Вибір конкретного статистичного показника (коефіцієнта кореляції, коефіцієнта детермінації тощо) і можливість його застосування в реалізації процесу кореляції можна визначити в ході аналізу вихідних даних та результатів обчислення такого показника. В даній роботі запропоновано використовувати операцію обчислення показника лінійної кореляції Пірсона з урахуванням факторів відносної схожості подій (w) і тимчасової затримки їх виникнення (dT). Для конкретизації та порівняння результатів обчислюються коефіцієнти кореляції між парами типів подій, оскільки показник відносної схожості залежить від типової структури подій.

В даному випадку важливість фактора схожості і тимчасової затримки для загального процесу кореляції не враховується.

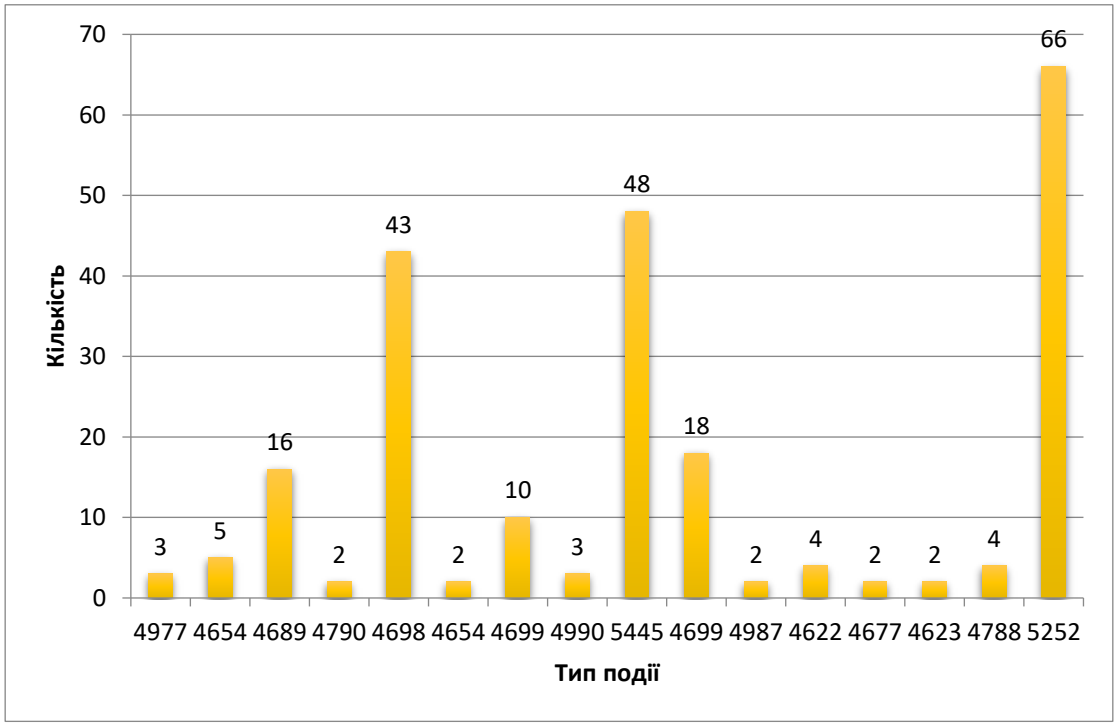


Рисунок 3.14 – Статистика типів подій, які зустрічаються рідко

Таблиця 3.1 – Значення для часу обробки вхідних наборів даних

Режим машинної обробки	Об’єм, Мб	Час, сек

локальний	9,3	3999
паралельний	9,3	555
паралельний	20,2	1220
паралельний	30,0	1476
паралельний	36,8	2022
паралельний	45,5	2790
паралельний	55,5	3288
паралельний	68,9	3666
паралельний	78,4	4290

Обчислені коефіцієнти кореляції можуть бути використані для попередньої оцінки процесів, що відбуваються в аналізованій інфраструктурі. Варто зазначити, що результати даних обчислень свідчать про наявність і характер лінійного зв'язку між зазначеними факторами, однак відсутність лінійного зв'язку не може свідчити про відсутність зв'язку як такового (можлива наявність складного нелінійного зв'язку).

Експериментальна оцінка оперативності реалізації процесу кореляції на розглянутому вище наборі даних була проведена з урахуванням двох сценаріїв на підставі рішення задачі пошуку загальних коефіцієнтів кореляції. У першому випадку використовувався однопоточний режим при частоті процесора 2 ГГц. При цьому обсяг вхідного набору даних дорівнював 9,3 МБ. У цьому випадку вважалося, що паралелізація обчислень для виконання процесу кореляції подій безпеки була відсутня. Результати, отримані в цьому випадку, розглядаються як контрольні, з якими порівнюються результати, отримані при реалізації паралельних обчислень.

У другому сценарії застосовувався метод паралелізації обчислень на основі поділу набору даних на частини. Обчислення проводилися на процесорі 8x2ГГц ЦПУ з реалізацією восьми потоків. В якості вхідного набору використовувався набір даних, збільшений в порівнянні з попереднім в 8 разів. Таким чином, максимальний обсяг вхідного набору даних досягав 78,4 МБ.

Результати експериментальної оцінки часу обробки вхідних наборів даних представлені в табл. 3.1.

Залежності часу обробки даних від обсягу вхідного набору даних подано в додатку 4 на графіку 1.

Аналізуючи дані додатку 4 на графіку 1 можна зробити наступні висновки. По-перше, час обробки даних при вирішенні задачі кореляції подій безпеки зменшується зі збільшенням кількості паралельних потоків. При цьому ця залежність близька до прямо пропорційної, тобто застосування восьми потоків дозволило скоротити час обробки приблизно у вісім разів. З іншого боку, залежність часу обробки даних від обсягу вхідного набору близька до лінійного виду. Так, якщо обсяг вхідного набору збільшити у вісім разів, то час обробки даних також збільшиться приблизно у вісім разів. Це є цілком закономірним результатом.

Hadoop має 4 різних алгоритми авто налаштування, ручне і пропорційне, і алгоритми для обчислення розподілу завдань по вузлах, кожен зі своїми власними параметрами конфігурації. Розподіл завдань вузлів виконується драйвером Hadoop. Ця робота є фактично основним фактором продуктивності каркаса. Він складається по суті з визначення того, скільки завдань, буде йти до кожного вузла для виконання, з набору завдань, які були надіслані клієнтським додатком. Кожен набір завдань, відправлених на вузол називається «пучком» і роллю балансування навантаження (або плануванням завдань) алгоритм для оптимізації продуктивності шляху регулювання кількості завдань, відправлене кожен вузол.

Додаток виконується на 2, 4 і 8 вузлах з використанням автоматичного налаштування і пропорційні алгоритми балансування навантаження. Результати

говорять, що вони залежать від числа вузлів і балансування навантаження алгоритму.

Таблиця 3.2 – Дані алгоритму пропорційного балансування навантаження

Номер вузла	Середній час виконання, (мс)	Середній вузол виконання, час (мс)	Середній транспорт, час (мс)	Середня черга, час (мс)
2	84,40	78,00	6,40	80,80
4	92,30	82,80	9,50	40,40
8	112,73	102,20	10,53	26,93

Таблиця 3.3 – Дані алгоритму авто налаштування балансування навантаження

Номер вузла	Середній час виконання, (мс)	Середній вузол виконання, час (мс)	Середній транспорт, час (мс)	Середня черга, час (мс)
2	92,93	85,57	7,37	20,30
4	104,64	96,44	8,20	18,68
8	112,00	102,55	9,45	20,20

Таблиця 3.2 і таблиця 3.3 показують, пропорційне і автоматичне завантаження даних налаштованих алгоритмом балансування на 2, 4 і 8 вузлах. В додатку 4 на Графіках 2 і 3 є графіками, які показують відносини між даними таблиці 4.2 і таблиці 4.3 відповідно.

Якщо цей додаток виконуються з використанням послідовного алгоритму, без використання Hadoop сіток на кластерах, то він займає близько 14-16 секунд, якщо алгоритм балансування навантаження використовується в кластерному середовищі, то він займає 3-4 секунди. Можна сказати, за допомогою Hadoop сітки на кластерному середовищі це паралельне прискорення і ефективність застосування максимально високі в порівнянні з послідовним впровадженням програми.

На основі системи, зроблено кілька експериментів. Деякі складні заяви SQL запитів, включаючи «де», «і», «або» були розроблені, щоб перевірити ефективність системи.

В додатку 5 графік 1 показує результати експерименту прикладу умови запиту з використанням п'яти вузлів. Відображається порівняння між послідовним методом і Hadoop паралельним методом.

Було встановлено, що під час запису мільйонних даних, час виконання різко зростає, а ефективність вилучення Hadoop є набагато більш ефективною, ніж послідовний метод на цьому рівні. Hadoop не показав свої переваги при розрахунку малої кількості даних. По-перше, вартість зв'язку між сервером і вузлами залежить від продуктивності мережі. По-друге, для виконання паралельних задач в тестовому середовищі велика кількість вузлів не доступні. По-третє, сервер не має можливості використовувати свої алгоритми балансування навантаження.

В додатку 5 графік 2 показує відносне прискорення, цільового запиту, що включає в себе 640000 записів.

Під середовищем Hadoop, побудувана розподілена паралельна система пошуку інформації. Експериментальні результати показують Hadoop, як платформу на якій не тільки легко побудувати паралельну систему розрахунку і зробити час вилучення коротше традиційного пошуку, але може також зменшити навантаження на сервер, і ефективно поліпшити ефективність.

Продемонструємо використання Hadoop по розпаралелюванню популяційного алгоритму вибору ознак.

Вимірювання для оцінки придатності:

$$f(s) = \frac{k\overline{r_{cf}}(s)}{\sqrt{k+k(k-1)\overline{r_{ff}}(s)}} \qquad (4.1)$$

де k загальне число функцій, $\overline{r_{cf}}(s)$ середня кореляційна функція класу $\overline{r_{ff}}(s)$ середня функція комбінації ознаки s. Оцінка є найбільш трудомісткою частиною

алгоритму і розпаралелювання. Результати класифікації по зниженню даних можна бачити у таблиці 3.4.

Таблиця 3.4 – Результати за класифікацією експериментів на різних наборах даних (2/3 навчання, 1/3 тесту)

Набір даних	Точність з вибором та (без вибору)	Кількість функцій	Особливість вибору
Armstrong	0,950 6 0,04 (0,946 6 0.02)	28,2 68,2	0,22%
Golub	0,929 6 0,05 (0,925 6 0.03)	29,4 64,7	0,41%
Shipp	0,924 6 0,07 (0,924 6 0,05)	28,4 62,1	0,40%
West	0,806 6 0,05 (0,775 6 0,09)	21,8 66,0	0,31%

Таблиця 3.5 – Час виконання паралельних обчислень функції придатності для набору даних West, використовуючи різні кількості робочих і процесорних ядер

Кількість робіт. (Число)	1 (10)	2 (20)	3 (30)	4 (40)	5 (50)	1 (1)
Середній час роботи (хв)	12,76	6,57	4,51	3,42	2,70	109,56
Прискорення	8,56	16,68	22,29	32,04	40,58	1,00

Результати збігаються з використанням всіх маркерів експресії генів, в той час як Generat значно знижено підпис (до 99,78%). Крім того, розраховується міра, яка добре масштабується з числом використовуваних ядер.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Платформа IDS на базі Hadoop по аналізу апостріорних даних системи	1. Управління подійними логами системи	Моніторинг, візуалізація та аналіз швидкодії масового збору даних з багатьох внутрішніх корпоративних джерел і зовнішніх джерел, таких як бази даних про уразливість;

		надання зведеного подання інформації, пов'язаної з безпекою; виконання аналізу потокових даних в режимі реального часу.
	2. Управління інформаційною безпекою	Підвищення інформаційної безпеки та ситуаційної обізнаності Виявлення аномалій і підозрілих дій, а також для зіставлення декількох джерел інформації в узгоджене уявлення
	3. Управління обчислювальними ресурсами	Інтелектуальне управління ресурсами кластеру Оптимізація часу аналізу логів системи на великих массивах данних

Таблиця 4.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	lucene	Teradata	Vertica			
1.	Вартість експлуатації	20 тис. грн/міс.	25 тис. грн/міс.	50 тис. грн/міс.	70 тис. грн./міс.	-	-	+
2.	Вартість обслуговування	10 тис. грн/міс.	15 тис. грн/міс.	40 тис. грн/міс.	30 тис. грн/міс.	-	-	+
3.	Наявність шифрування	Присутня	Присутня	Відсутня	Відсутня	-	+	-

№ п/п	Техніко-економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	lucene	Teradata	Vertica			
4.	Інтелектуальне балансування навантаження	Присутнє	Присутнє	Присутнє	Відсутнє	-	-	+
5.	Забезпечення високої доступності	Присутнє	Присутнє	Відсутнє	Відсутнє	-	+	-
6	Підтримка Docker	Відсутній	Відсутній	Присутній	Відсутній	+	-	-
7.	Наявність консольного інтерфейсу	Присутній	Присутній	Присутній	Присутній	-	+	-

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Data Platform	Syncfusion	Наявні	Доступні
2.		hortonworks	Наявні	Доступні
3.		cloudera	Наявні	Недоступні
Обрана технологія реалізації ідеї проекту: Syncfusion.				

4.	Інструменти Big Data	Hadoop	Наявні	Доступні
5.		Spark	Наявні	Доступні
6.		Storm	Наявні	Недоступні
Обрана технологія реалізації ідеї проекту: Hadoop.				
4.	Мова програмної реалізації	Компільована у машинний код (C, C++)	Наявні	Доступні
5.		Компільована у байткод (Java, C#)	Наявні	Доступні
6.		Скриптова/інтерпретована (Python, Perl, Lua, Ruby)	Наявні	Доступні
Обрана технологія реалізації ідеї проекту: Компільована у байткод (Java, C#)				

4.3 Аналіз ринкових можливостей запуску стартап-проетку

Таблиця 4.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	5
2.	Загальний обсяг продаж, грн/ум.од	1000 грн/ум. од.
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	Недискримінаційні якісні
5.	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6.	Середня норма рентабельності в галузі (або по ринку), %	70%

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару

1.	Управління подійними логами системи	1. Малий бізнес 2. Середній бізнес	Управління та моніторинг об’єму інформації про мережевий трафік логів систем	Підвищення адаптивності бізнесу Підвищення швикості надання інфраструктури Зменшення часу розгортання
2.	Управління інформаційною безпекою	1. Малий бізнес 2. Середній бізнес	Наявність вимог до високої швидкості виявлення аномалій	Можливість гнучкого налаштування Конфігурація параметрів доступності Конфігурація стратегії відмовостійкості
3.	Оптимізація ресурсів	1. Малий бізнес 2. Середній бізнес	Різні об’єми та характеристики ресурсів	Швидка ініціалізація та конфігурація ресурсів Підвищення коефіцієнту використання ресурсів

Таблиця 4.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Крадіжка інтелектуальної власності	Крадіжка ідеї або ключової інтелектуальної інновації	Відсудження прав інтелектуальної власності Забезпечення якіснішого захисту інформації Зміна методики шифрування приватного ключа Попередження користувачів із подальшою співпрацею для мінімізації фактор загрози
2.	Отримання несанкціонованого доступу сторонніми особами	Хакерська атака що може призвести до компроментації даних клієнтів	Залучення спеціалістів з інформаційної безпеки Використання засобів шифрування та резервного копіювання

3.	Відсутність ринку	Відсутність шляху збуту товару внаслідок помилкового орієнтування	Ретельний розгляд проблем потенційних клієнтів Залучення експертів та менторів Консультації із спеціалістами
4.	Недостача капіталовкладень	Витрачені усі кошти до моменту виходу на ринок	Пошук нових джерел інвестицій

Таблиця 4.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Отримання інвестицій	Отримання капіталу що необхідний для реалізації продукту	Розробка продукту
2.	Успішна маркетингова політика	В результаті проведеної маркетингової політики отримана висока зацікавленість користувачів	Підтримка стабільної роботи системи та проведення масштабування системи Збільшення цін на використання сервісу Використання подібної маркетингової стратегії надалі для залучення нових користувачів
3.	Поглинання конкурентами	Пропозиція купівлі проекту або розроблених технологій одним із конкурентів	Розвиток розроблених технологій Оцінка вартості розроблених технологій

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Олігополія	Незначна кількість конкурентів Велика ринкова сила Схожість використовуваних технологій	Інформування ринку щодо появи нової платформи управління подійними логами системи

Галузевий	Загроза появи нових конкурентів Виркова влада споживачів Висока потреба у товарі	Інформування ринку щодо якості використовуваної новаторської технології Пропозиція гнучких цін
Внутрішньогалузева	Діяльність в одній галузі економіки Надання сервісів одного типу	Зменшення вартості сервісу Примноження каналів розподілу
Товарно-видова	Надання різних сервісів одного типу	Маркетингова політика
Цінова	Використання цін для покращення економічних умов збуту	Зменшення вартості платформи Використання нових каналів розподілу
Марочна	Пропозиція схожої платформи Спільна цільова аудиторія	Інформування ринку щодо появи нової платформи управління подійними логами системи

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	“Lucene”, “Teradata”, “Verticas”	Розмір капіталовкладень, Забезпечення гнучких цін, Доступ до каналів розподілу, Витрати на масштабах	Відсутні	Змінні витрати: Виробничі непрямі дегресивні - Системи інформації: пропаганда, реклама та директ-маркетинг, - Рівень чутливості до цін: споживачі орієнтовані на цінність продукту - Продуктова диференціація: якість, спосіб отримання сервісу,	Копіювання функціоналу, Монополізація дистриб’юторів, Демпінгування

				швидкість обслуговування Методи контролю якості: тестування та профілювання, прототипування, інспектування коду, аналіз архітектури програмного забезпечення	
Висновки	CR4 = 92% Індекс Херфіндаля-Хіршмана (HHI) = 6565 Значення показників вказує на високу концентрацію (монополізацію) даного ринку	Можливості входу на ринок забезпечить мінімізація цін, швидкість та простота надавання послуги споживачам і співпраця із головними гравцями ринку. В результаті аналізу проектів на народно-громадських інтернет-платформах потенційних конкурентів знайдено не було	Відсутні	Клієнти диктують умови гнучкості цінової політики, високої і довгострокової якості послуг та наявність кооперації із сервісами, що вони використовують	Пропонування вигідних умов дистриб'юторам, забезпечення захисту інтелектуальної власності, гнучкість цінової політики

Таблиця 4.9. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Унікальність сервісу	Розроблений продукт має унікальне співвідношення ціна / якість для свого цінового діапазону

2.	Цінова політика	Отримання прибутку здійснюється за рахунок гнучкої моделі оплати
3.	Модель “бізнес для бізнесу”	Бізнес модель грунтується на співпраці із іншими платформами управління інфраструктурами. Даний підхід дозволить обійти цінову конкуренцію на ринку цільової аудиторії

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею.

Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Таблиця 4.10. Порівняльний аналіз сильних та слабких сторін «Adimas»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з Adimas						
			–3	–2	–1	0	+1	+2	+3
1.	Унікальність сервісу	14						+	
2.	Цінова політика	19							+
3.	Модель “бізнес для бізнесу”	13					+		

Таблиця 4.11. SWOT- аналіз стартап-проекту

Сильні сторони: Якість та довготривалість Низькі ціни	Слабкі сторони: Нестача капіталовкладень Бізнес-модель залежить від політики окремих бізнесів
Можливості: Інвестиції	Загрози: Крадіжка інтелектуальної власності

Реалізація бізнес-моделі Розширений функціонал Висока зацікавленість цільової аудиторії	Компроментація даних клієнтів Відсутність ринку

Таблиця 4.12. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Розробка власних засобів віртуалізації	Ймовірне	12 місяців
2.	Маркетингова кампанія для приваблювання користувачів	Малоймовірне	2 місяці
3.	Пропонування безкоштовних тарифів	Малоймовірне	1 місяць
4.	Пошук бізнесів інших галузей для співпраці	Дуже ймовірне	6 місяців
Обрана альтернатива: Пошук бізнесів іншої галузі для співпраці			

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.13. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Адміністратори інфраструктур безпеки малого бізнесу	Висока	65%	Середня	Низькі бар’єри входу
2.	ІТ-підрозділи середнього бізнесу	Висока	78%	Середня	Низькі бар’єри входу
3.	Власники платформ по обробці подійних логів	Мала	35%	Середня	Високі бар’єри входу
Які цільові групи обрано: адміністратори інфраструктур безпеки, малого бізнесу, ІТ-підрозділи середнього бізнесу					

Відповідно до проведеного аналізу можна зробити висновок, що підходящою цільовою групою для розповсюдження даного програмного продукту є працівники інфраструктур безпеки, малого бізнесу, IT-підрозділи середнього бізнесу в цілому та будь-які підприємства котрі використовують системи IDS . Відповідно до стратегії охоплення ринку збуту товару обрано стратегію масового маркетингу, оскільки для підприємств, IT працівників та IT компаній у цілому надається стандартизований продукт з можливістю розширення функціональності за домовленістю (відповідно до ліцензії).

Таблиця 4.14. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Надання платформи малому та середньому бізнесу	Вибірковий розподіл	Здатність протистояти прямим конкурентам Низькі витрати Ефективна співпраця	Стратегія диференціації

Таблиця 4.15. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні	Забирати та залучати нових	Веб-інтерфейс керування інфраструктурою Інтелектуальних розподіл обчислювальних ресурсів	Стратегія лідера. Розширення первинного попиту

Таблиця 4.16. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Відповідність затвердженим характеристикам Висока ступінь надійності системи Простий інтерфейс адміністратора Гнучка цінова політика Оперативна підтримка продукту	Стратегія диференціації	Формування регулярного попиту Збільшення разового використання послуги Виявлення нових груп споживачів Нові напрями застосування існуючої послуги	Інноваційність технології Низькі ціни Простота використання

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку – стратегію диференціації, як базову стратегію конкурентної поведінки – стратегію заняття конкурентної ніші.

4.5. Розроблення маркетингової програми стартап-проекту

Таблиця 4.17. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Управління подійними логами системи	Спрощення бізнес-процесів, підвищення адаптивності бізнесу	Якість надання послуг Інноваційність технологій що використовуються Простота використання Цінова перевага
2	Управління інформаційною безпекою	Реалізація відмовостійкості та високодоступності застосунків	Якість надання послуг Інноваційність технологій що використовуються Простота використання Цінова перевага

3	Управління обчислювальними ресурсами	Швидка ініціалізація та конфігурація ресурсів Підвищення використання ресурсів	Якість надання послуг Інноваційність технологій що використовуються Простота використання Цінова перевага

Таблиця 4.18. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Програмний продукт що надає можливість пришвидшити час обробки апостріорних даних системи		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Кількість		1 шт.
	Якість: стандарти якості постачання програмних продуктів		
	Пакування: комп'ютерна дискета		
	Марка: Petruchio IDS Management Platform		
III. Товар із підкріпленням	Програмний продукт		
	Програмний продукт, технічна підтримка та підписка на оновлення		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності			

Таблиця 4.19. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	5 тис. грн. – 10 тис. грн	5 тис. грн. – 100 тис. грн	20 000 грн./міс.	1 -5 тис. грн

Таблиця 4.20. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Закупівля здійснюється через довірені джерела	Інформування користувачів Доступ користування	Канал одного рівня	Селективна з використанням комбінованого

		сервісом		каналу збуту

Таблиця 4.22. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Автоматизація бізнес-процесів Вимоги до високодоступності та відмовостійкості	Прямі офіційні	Послідовність в реалізації обраної позиції Доступність та об'єктивність інформації про фірму і товар Унікальність послуги	Формування у цільової аудиторії обізнаності про появу нового продукту Інформування користувачів про властивості та переваги продукту Інформування користувачів про нові способи використання відомого продукту Пояснення цільовій аудиторії принципу роботи платформи Виправити у користувачів неправильні представлення про продукт	Раціоналістична стратегія реклами

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

Висновки по розділу

В четвертому розділі описано стратегії та підходи з розроблення стартап-проекту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як

висновок було вказано що існує можливість ринкової комерціалізації проекту. Розглянувши потенційні групи клієнтів, бар’єри входження, стан конкуренції та конкурентоспроможність проекту було встановлено що проект є перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проекту та доведено доцільність подальшої імплементації проекту.

ВИСНОВКИ

У рамках даної роботи здійснено дослідження системи Data Mining і машинних технік навчання для виявлення вторгнення в кібербезпеку робототехнічних і автономних систем.

Основу методів Data Mining становлять всілякі методи класифікації, моделювання і прогнозування. До методів Data Mining нерідко відносять статистичні методи (дескриптивний аналіз, кореляційний і регресійний аналіз, факторний аналіз, дисперсійний аналіз, компонентний аналіз, дискримінантний аналіз, аналіз часових рядів). Такі методи, проте, припускають деякі апріорні уявлення щодо аналізованих даних, що виникає певна розбіжність з цілями Data Mining (виявлення раніше невідомих нетривіальних і практично корисних знань).

Одне з найважливіших призначень методів Data Mining полягає в наочному поданні результатів обчислень, що дозволяє використовувати інструментарій Data Mining людьми, які не мають спеціальної математичної підготовки. У той же час, застосування статистичних методів аналізу даних вимагає доброго володіння теорією ймовірностей і математичної статистики.

Оцінка існуючих вразливостей в сенсорах робототехнічних систем базується на дослідженні вразливостей екстероцептивного датчику, камери, мікрофону і т.д. Загалом роботи володіють цілим рядом вразливостей, серед яких незахищені канали зв'язку, проблеми з аутентифікацією, слабка криптографія і відсутність авторизації.

До головних атак, що є можливими у спектрі роботи роботів можна віднести такі атаки, як: Stealth attack, Replay attack, Covert attack, False-Data injection, DoS attack, Remote access, Eavesdropping.

В даний час засоби виявлення, попередження і запобігання комп'ютерних атак і шкідливої активності, а також моніторингу та управління безпекою представлені різними класами рішень. Одним з таких класів є системи SIEM (Security Information and Event Management).

Основними завданнями SIEM-систем є збір великих масивів гетерогенних даних про події безпеки і виявлення інцидентів і загроз безпеці в результаті їх обробки. При цьому однією з достатніх гострих проблем, що стоять перед сучасними SIEM-системами, є проблема обробки великих даних, яка викликана необхідністю обробки величезних масивів різнорідних даних про події безпеки (логів), що надходять в SIEM- систему від різних джерел. У якості джерел великих даних виступають операційні системи, системи управління базами даних, антивірусні засоби, мережні елементи, системи виявлення атак і т.д.

Проведений аналіз впливу архітектурних тактик на Security Analytics Systems показав загальну оцінку впливу тактик на атрибути точності, надійності, продуктивності, масштабування. Так наприклад тактика об'єднання декількох методів виявлення і тактика об'єднання виявлення на основі сигнатур і аномалій доповнюють один одного для досягнення якісного атрибуту точності, однак об'єднання обох буде збільшувати час відгуку. Таким чином, архітектор може розглянути можливість видалення будь-якого з них в залежності від вимоги. Огляд дозволив стверджувати, що, ймовірно, будуть існувати більш глибокі залежності між

ідентифікованими архітектурними тактиками; ще одна область для емпіричних досліджень архітектурної тактики для систем аналітики безпеки.

Відомо кілька засобів реалізації паралельних обчислень. До найбільш поширених належать Hadoop і Spark. При цьому Spark на великих обсягах вхідних даних показує більш високу продуктивність. У той же час Spark є більш молодим засобом.

Основні цілі роботи:

- формалізація задач виявлення зв'язків між типами подій безпеки і оцінки залежності сили зв'язків від розподілу подій у часі, що вирішуються в SIEM-системах при кореляції подій безпеки;
- реалізація алгоритмів вирішення цих завдань в Hadoop;
- експериментальна оцінка отриманих рішень.

Для проведення експериментів використовувалася платформа Syncfusion v3.2.0 з двома процесорами Intel CORE I5 @ 2.1 ГГц та операційною системою Windows 10. Машині було виділено 4 thread, зарезервовано (reserved) 2 ГГц процесора, також було зарезервовано 16 ГБ ОЗУ.

В роботі для проведення експериментальної оцінки в цілях підвищення достовірності отриманих результатів використовується вхідний потік даних, зібраний з комп'ютера, що зберігає дані аналізу кібервторгнень на робо технічні та автономні системи. В якості вхідних даних був взятий файл розміром 7 ГБ, в якому містилося приблизно 7 мільйонів подій, що відносяться до 81 типу. Вихідні дані були попередньо зібрані з журналу подій безпеки ОС Windows 10 на офісному комп'ютері за період, приблизно рівний 2 діб.

Якщо цей додаток виконуються з використанням послідовного алгоритму, без використання Hadoop сіток на кластерах, то він займає близько 14-16 секунд, якщо алгоритм балансування навантаження використовується в кластерному середовищі, то він займає 3-4 секунди. Можна сказати, за допомогою Hadoop сітки на кластерному середовищі це паралельне прискорення і ефективність застосування максимально високі в порівнянні з послідовним впровадженням програми.

Було встановлено, що під час запису мільйонних даних, час виконання різко зростає, а ефективність вилучення Hadoop є набагато більш ефективною, ніж послідовний метод на цьому рівні. Hadoop не показав свої переваги при розрахунку малої кількості даних. По-перше, вартість зв'язку між сервером і вузлами залежить від продуктивності мережі. По-друге, для виконання паралельних задач в тестовому середовищі велика кількість вузлів не доступні. По-третє, сервер не має можливості використовувати свої алгоритми балансування навантаження.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ullah, F., and Babar, M.A Architectural Tactics for Big Data Cybersecurity Analytic Systems: A Review, 2018 - 48
2. Антамошин А.Н., Близнова О.В., Бобов А.В., Большаков А.А., Лобанов В.В., Кузнецова И.Н. Интеллектуальные системы управления организационно-техническими системами. – М.: Горячая линия – Телеком, 2006. – 160 с.
3. Бодров О.А., Медведев Р.Е. Предметно-ориентированные экономические информационные системы. – М.: Горячая линия – Телеком, 2013. – 244 с.
4. Бородакий Ю. В., Лободинский Ю. Г. Эволюция информационных систем (современное состояние и перспективы). – М.: Горячая линия – Телеком, 2011. – 368 с.
5. Васильев Р.Б., Калянов Г.Н., Лёвочкина Г.А. Управление развитием информационных систем. – М.: Горячая линия – Телеком, 2009. – 368 с.
6. Когаловский М. Р. Перспективные технологии информационных систем. – М.: ДМК Пресс; Компания АйТи, 2003. – 288 с.
7. Когаловский М. Р. Энциклопедия технологий баз данных. – М.: Финансы и статистика, 2002. – 800 с.
8. Л. Л. Винокуров, Д. В. Леонтьев, А. Ф. Гершельман. СУБД ADABAS – основа универсального сервера баз данных // СУБД. – 1997. – №2. – С. 36-40.
9. Сахаров А. А. Концепции построения и реализации информационных систем, ориентированных на анализ данных // СУБД. – 1996. – № 4. – С. 55-70.

10. Ульман Дж. Основи систем баз даних. – М.: Фінанси і статистика, 1983. – 334 с.

11. Чаудхари С. Методи оптимізації запитів в реляційних системах / / Системи управління базами даних. – М., 1998. – № 3. – С. 22-36.

12. D. Hackathorn. Reinventing Enterprise Systems Via Data Warehousing. – Washington, DC: The Data Warehousing Institute Annual Conference, 1995.

13. E. F. Codd, S. B. Codd, C. T. Salley. Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate. – E. F. Codd & Associates, 1993.

14. EP Harris, K. Ramamohanarao . Join algorithm costs revisited / / The VLDB Journal. – 1996. – Vol. 5, № 1. – P. 64 - 84.

15. G. Gardarin, F. Sha, and Z.-H. Tang. Calibrating the query optimizer cost model of IRO-DB, an objectoriented federated database system / / Proceedings of 22th International Conference on Very Large Data Bases (VLDB'96), September 3-6, 1996, Mumbai (Bombay), India. – P. 378-389.

16. Georges Gardarin, Jean-Robert Gruser, Zhao-Hui Tang . Cost-based Selection of Path Expression Processing Algorithms in Object-Oriented Databases / / Proceedings of 22th International Conference on Very Large Data Bases (VLDB'96), September 3-6, 1996, Mumbai (Bombay), India. – P. 390-401.

17. Goe tz Graefe . Query Evaluation Techniques for Large Databases / / ACM Computing Surveys. – 1993. – Vol. 25, № 2. – P. 73-170.

18. J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, H. Pirahesh. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-

Totals // Data Mining and Knowledge Discovery. – 1997. – № 1. – P. 29-53.

19. K. Parsaye. New Realms of Analysis: Surveying Decision Support // Database Programming & Design. – 1996. – № 4. – P. 26-33.

20. Mishra P., Eich MH . Join Processing in relational databases. / / ACM Computing Surveys. – 1992. – Vol. 24, № 1.

21. N. Raden. Данные, Данные и только данные // ComputerWeek-Москва. – 1996. – №8. – С. 28.

22. W. H. Inmon. Building The Data Warehouse (Second Edition). – NY, NY: John Wiley. – 1993.

23. Weimin Du , Ravi Krishnamurthy , Ming - Chien Shan . Query Optimization in a Heterogeneous DBMS / / Proceedings of 18th International Conference on Very Large Data Bases, August 23-27, 1992, Vancouver, Canada. – P. 277-291.

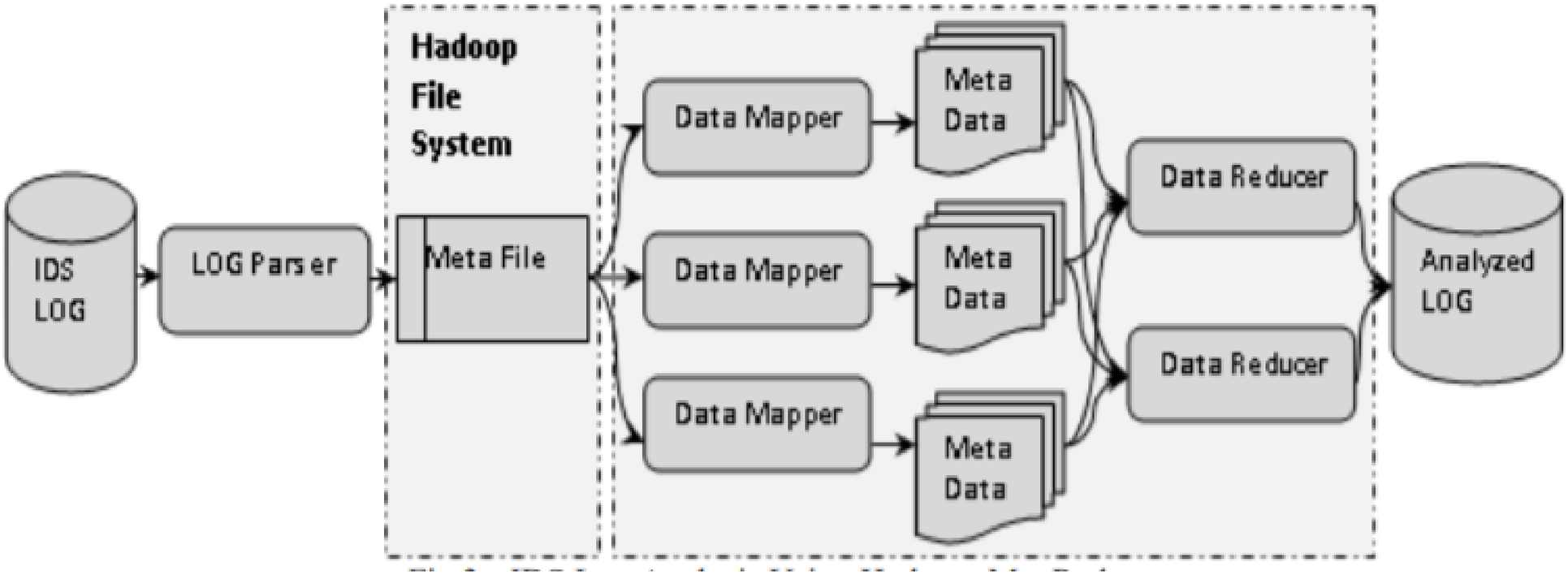
24. William S. Davis, David C. Yen The Information System Consultant's Handbook. Systems Analysis and Design. – CRC Press, 1998. – 800 с.

ДОДАТКИ

ДОДАТОК А

Загальна схема системи IDS з використанням Hadoop MapReduce

Загальна схема системи IDS з використанням Hadoop MapReduce



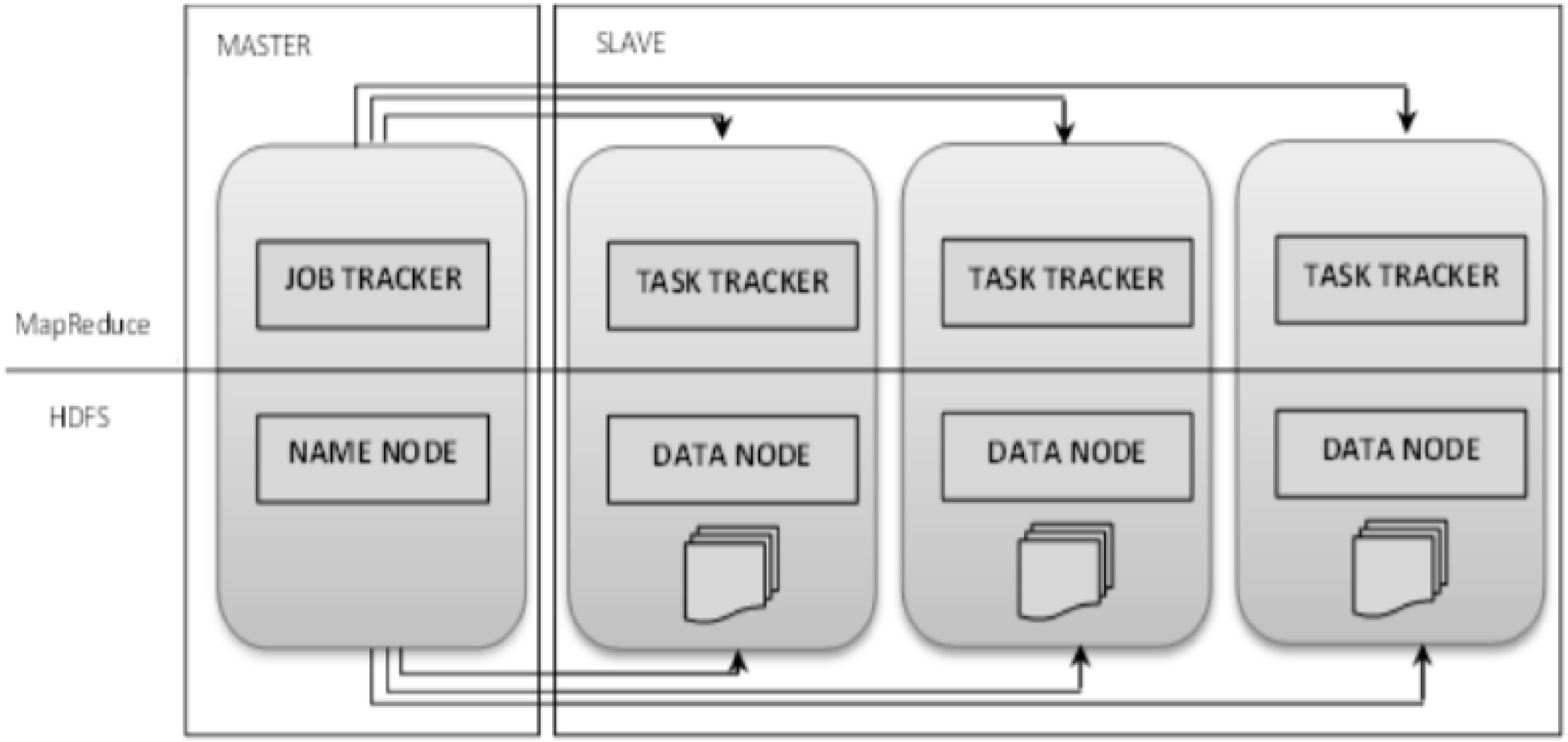
Демонстраційний плакат №_1_
до магістерської дисертації на тему
„Data Mining та машинні техніки навчання для виявлення вторгнення в кібербезпеку робототехнічних та автономних систем”

Розробив: __Петрухно І.Р.___
Прийняв: ___Бурлаков В.М.__

ДОДАТОК Б

Загальна схема Структура кластера Hadoop

Загальна схема Структура кластера Hadoop



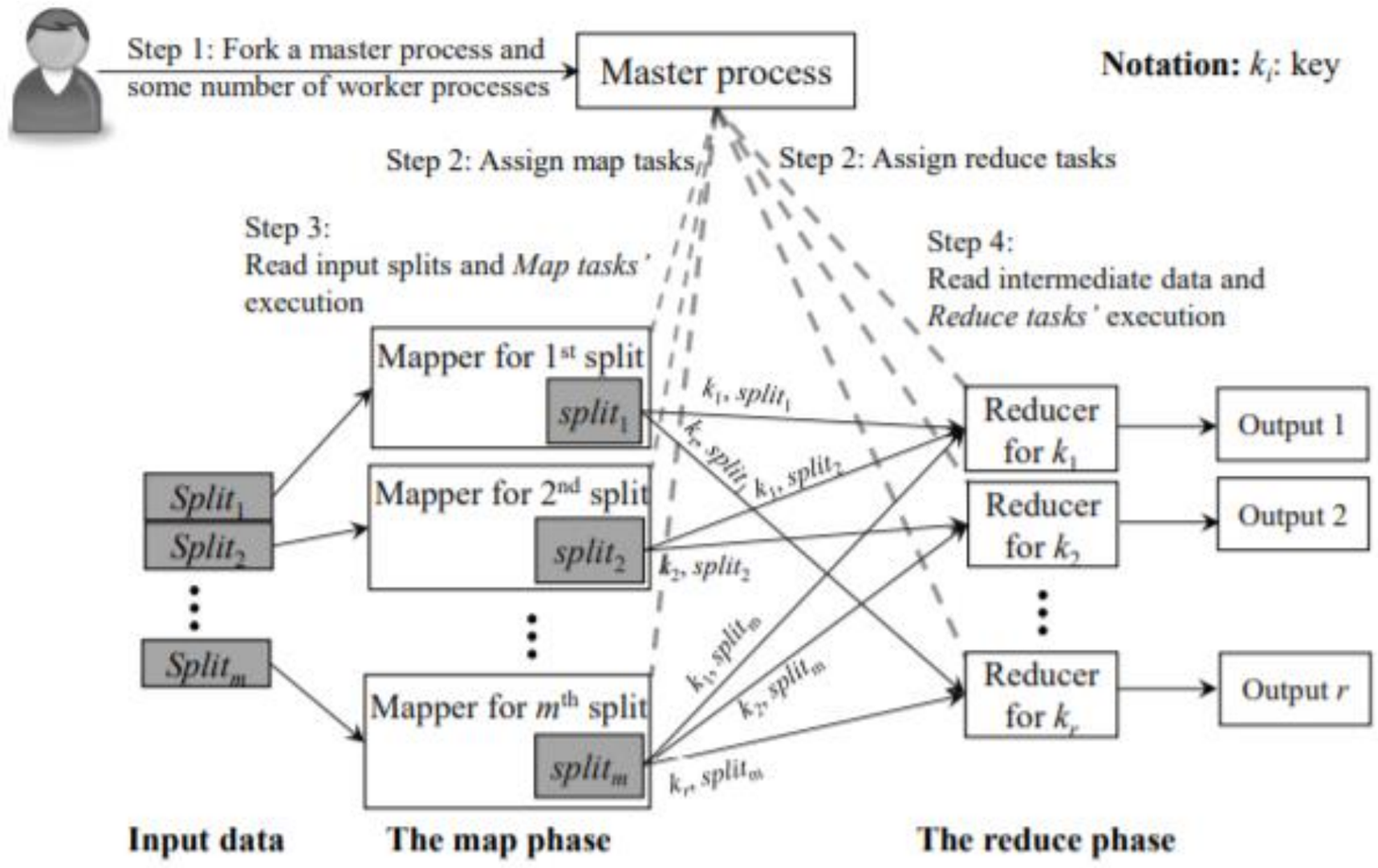
Демонстраційний плакат №_2_
до магістерської дисертації на тему
„Data Mining та машинні техніки навчання для виявлення вторгнення в кібербезпеку робототехнічних та автономних систем”

Розробив: __Петрухно І.Р.___
Прийняв: ___Бурлаков В.М.__

ДОДАТОК В

Алгоритм загального виконання MapReduce

Алгоритм загального виконання MapReduce



Демонстраційний плакат №_3_
до магістерської дисертації на тему
„Data Mining та машинні техніки навчання для виявлення вторгнення в кібербезпеку робототехнічних та автономних систем”

Розробив: __Петрухно І.Р.___
Прийняв: ___Бурлаков В.М.__

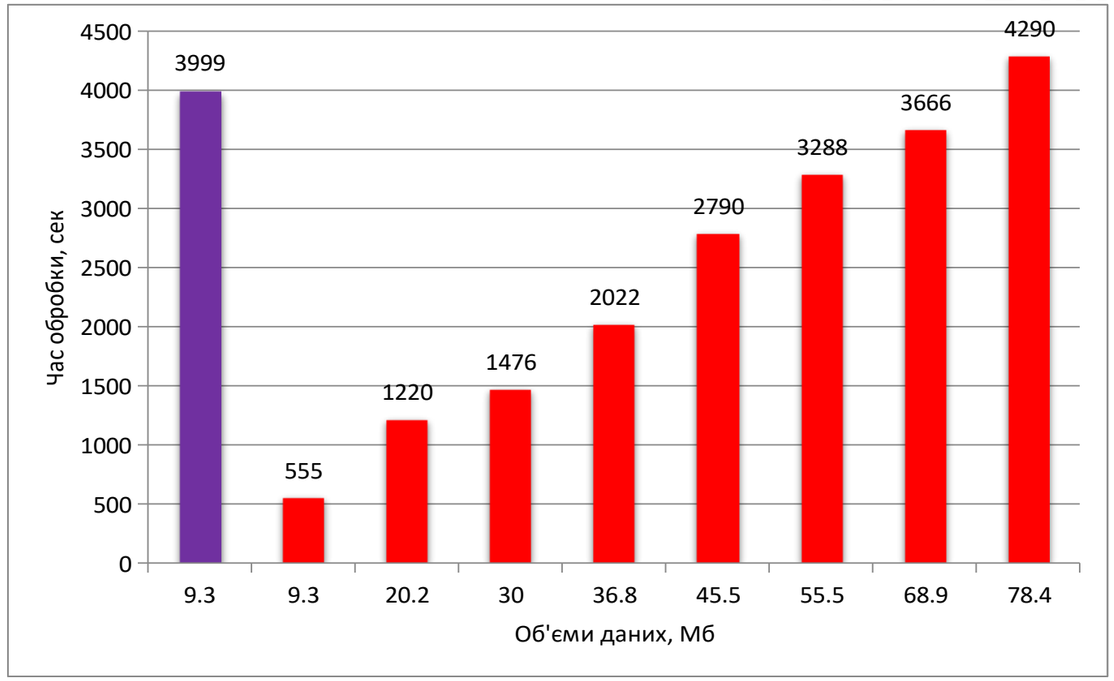
ДОДАТОК Г

Результати експериментів

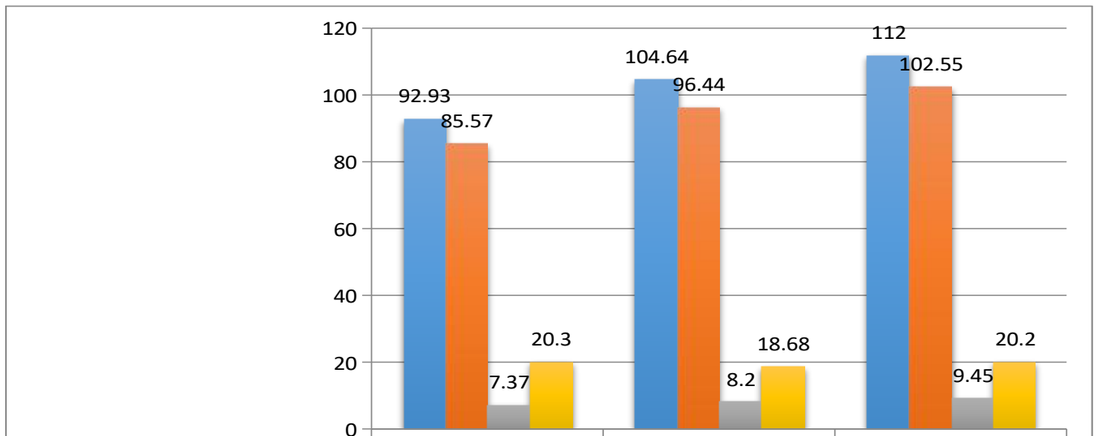
Час виконання

Результати експериментів

Час виконання



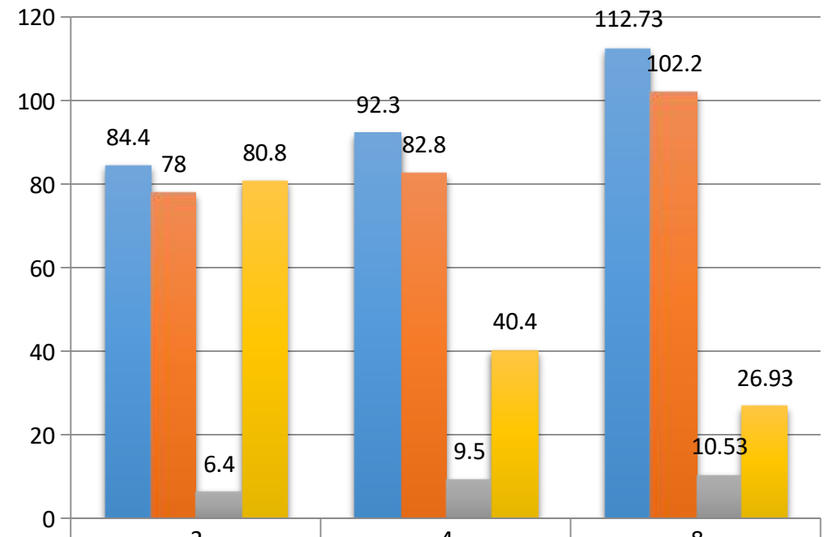
Графік 1 - Залежності часу обробки даних



	2	4	8
Середній час виконання, (мс)	92.93	104.64	112
Середній вузол виконання, час (мс)	85.57	96.44	102.55
Середній транспорт, час (мс)	7.37	8.2	9.45
Середня черга, час (мс)	20.3	18.68	20.2

Середній час виконання, (мс) — Середній вузол виконання, час (мс)
Середній транспорт, час (мс) — Середня черга, час (мс)

Графік 3 - Автоматичне налаштування балансування навантаження даних алгоритму



	2	4	8
Середній час виконання, (мс)	84.4	92.3	112.73
Середній вузол виконання, час (мс)	78	82.8	102.2
Середній транспорт, час (мс)	6.4	9.5	10.53
Середня черга, час (мс)	80.8	40.4	26.93

Середній час виконання, (мс) — Середній вузол виконання, час (мс)
Середній транспорт, час (мс) — Середня черга, час (мс)

Графік 2 - Пропорційне балансування навантаження даних алгоритму

Демонстраційний плакат №_4_
до магістерської дисертації на тему
„Data Mining та машинні техніки навчання для виявлення вторгнення в кібербезпеку робототехнічних та автономних систем”

Розробив: __Петрухно І.Р.___
Прийняв: ___Бурлаков В.М.__

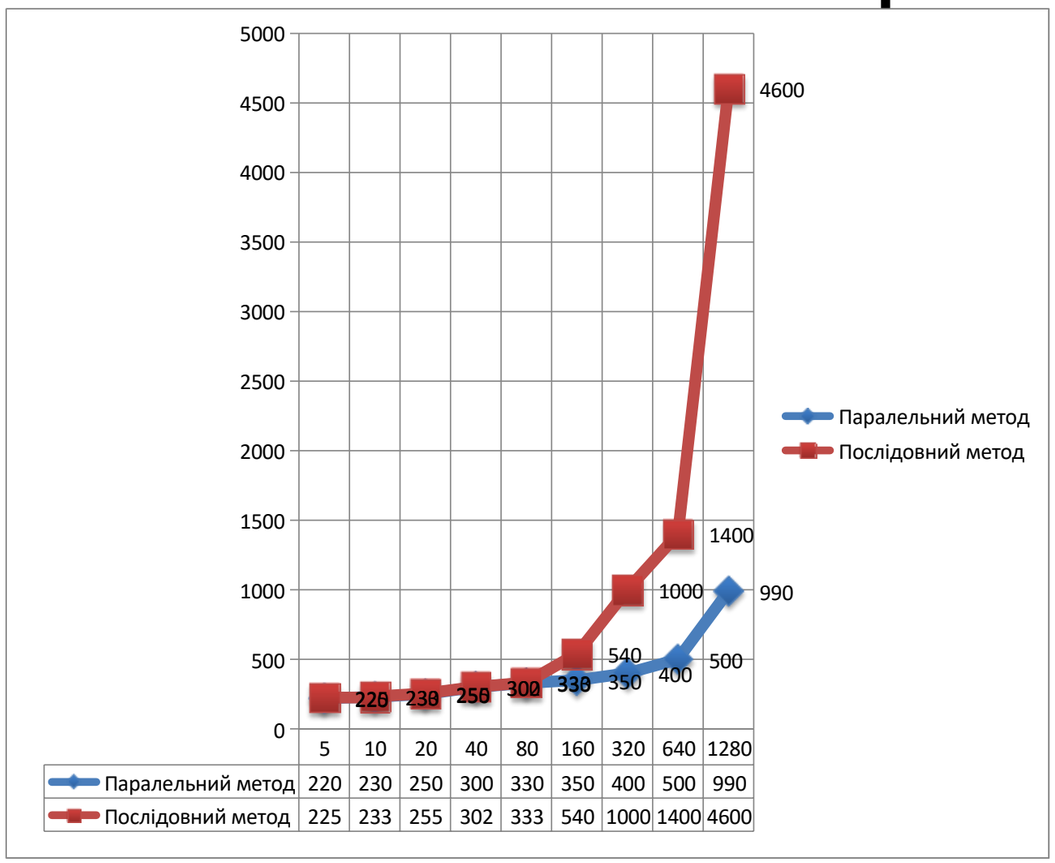
ДОДАТОК Д

Результати експериментів

Ефективність

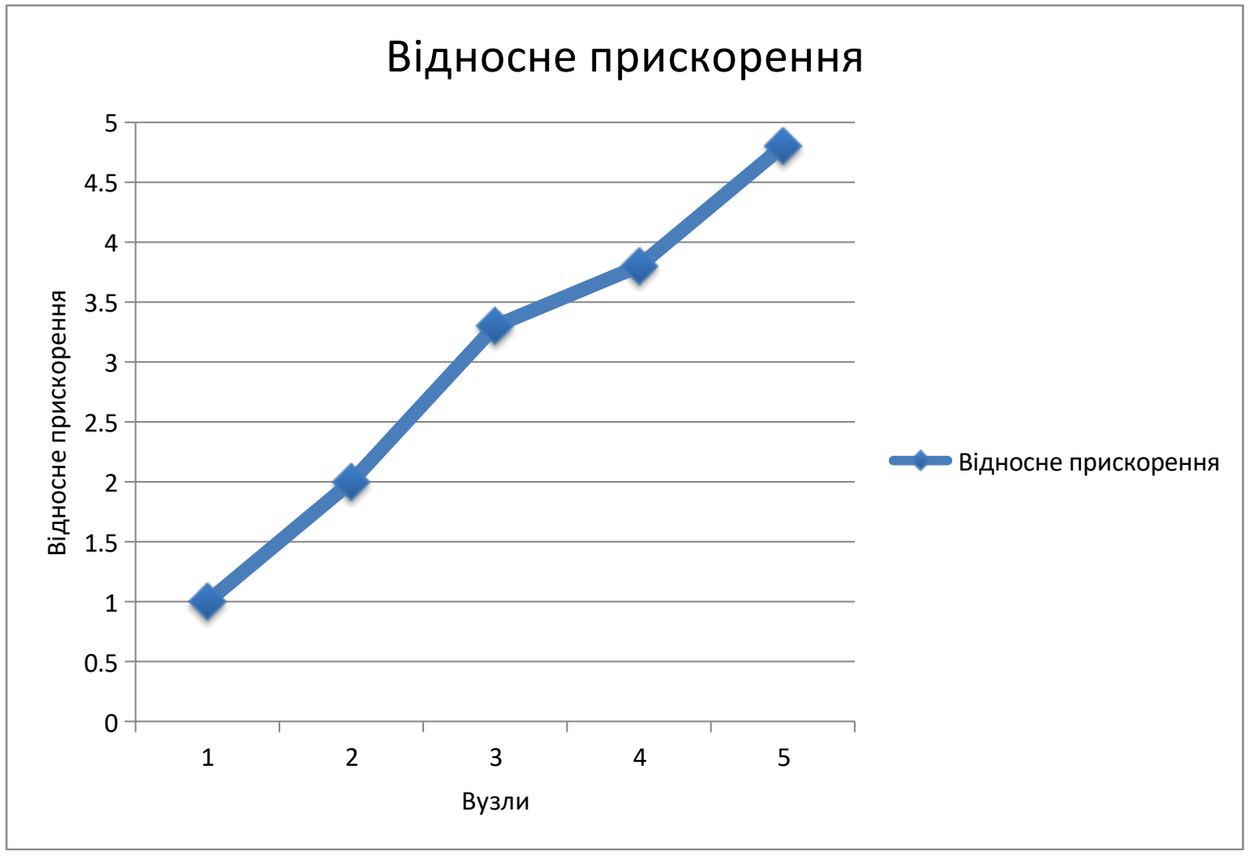
Результати експериментів

Ефективність

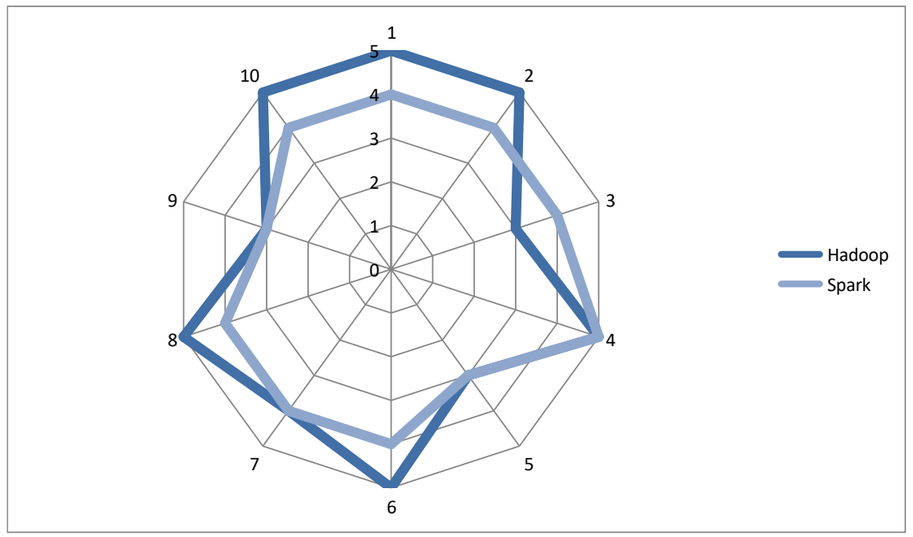


	5	10	20	40	80	160	320	640	1280
Паралельний метод	220	230	250	300	330	350	400	500	990
Послідовний метод	225	233	255	302	333	540	1000	1400	4600

Графік 1 - Результати експерименту послідовних і паралельних методів



Графік 2 - Відносне прискорення (640000 записів)



Графік 3 - Графік ефективності систем

Демонстраційний плакат №_5_
до магістерської дисертації на тему
„Data Mining та машинні техніки навчання для виявлення вторгнення в кібербезпеку робототехнічних та автономних систем”

Розробив: __Петрухно І.Р.___
Прийняв: ___Бурлаков В.М.__